

AD-A106 064

JET PROPULSION LAB PASADENA CA

F/G 22/2

ASSESSMENT OF AUTONOMOUS OPTIONS FOR THE DSCS III SATELLITE SYS--ETC(U)

AUG 81 D L PIVIROTTI, M MARCUCCI

NAS7-100

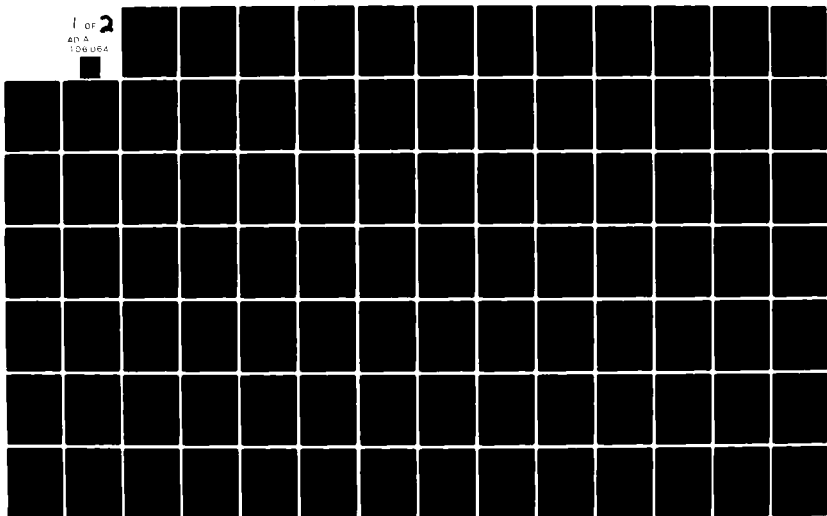
UNCLASSIFIED

JPL-7030-2-VOL-3

SD-TR-81-87-VOL-3

NL

1 OF 2
40 A
106 064



SD-TR-81-87

LEVEL III

①

AD A106064

Autonomous Spacecraft Project
Assessment of Autonomous Options
for the DSCS III Satellite System
Volume III: Options for Increasing the Autonomy
of the DSCS III Satellite

Approved for Public Release; Distribution Unlimited

DTIC
ELECTE
OCT 23 1981
S D E

6 August 1981

Interim Report for Period

1 November 1980 through 6 August 1981

Prepared for

U.S. Air Force Systems Command

Headquarters, Space Division

Through an agreement with

National Aeronautics and Space Administration

by

Jet Propulsion Laboratory

California Institute of Technology

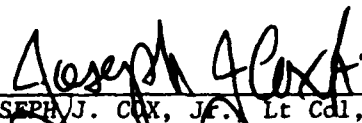
Pasadena, California 91109

81 10 22

This interim report was submitted by the Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California, 91109, under Contract NAS7-100, JPL Task Plan No. 80-1487, with the Headquarters, Space Division, Los Angeles AFS, California, 90009. Major Ralph R. Gajewski (YLXT) was the project officer. This report has been reviewed and cleared for open publication and/or public release by the appropriate Public Affairs Office (PAS) in accordance with AFR 190-17 and DODD 5230.9. There is no objection to unlimited distribution of this report to the public at large, or by DTIC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.


RALPH R. GAJEWSKI, Major, USAF
Chief, Advanced Materials
and Structures


JOSEPH J. COX, Jr., Lt Col, USAF
Chief, Advanced Technology Division


BURTON H. HOLADAY, Colonel, USAF
Director of Space Systems Planning
Deputy for Technology

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(18) 50

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER <u>19 TR-84-87-VOL-3</u>	2. GOVT ACCESSION NO. <u>AD-A206</u>	3. RECIPIENT'S CATALOG NUMBER <u>364</u>
4. TITLE (and Subtitle) <u>ASSESSMENT OF AUTONOMOUS OPTIONS FOR THE DSCS III SATELLITE SYSTEM.</u> <u>VOLUME III: OPTIONS FOR INCREASING THE AUTONOMY OF THE DSCS III SATELLITE.</u>		5. TYPE OF REPORT & PERIOD COVERED <u>Interim (1 Nov 80-6 Aug 81)</u>
7. AUTHOR(s) <u>Donna L. S./Pivirotto</u> <u>Michael/Marcucci</u>		6. PERFORMING ORG. REPORT NUMBER <u>-7030-2-VOL-3</u>
9. PERFORMING ORGANIZATION NAME AND ADDRESS <u>Jet Propulsion Laboratory</u> <u>California Institute of Technology</u> <u>4800 Oak Grove Dr., Pasadena, CA 91109</u>		8. CONTRACT OR GRANT NUMBER(s) <u>NAS7-100</u> <u>JPL Task Plan No. 80-1487</u>
11. CONTROLLING OFFICE NAME AND ADDRESS <u>HQ Space Division/YLXT</u> <u>Box 92960 Worldway Postal Center</u> <u>Los Angeles CA 90009</u>		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <u>12. 184</u>
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) <u>(9) Interim Rept.</u> <u>4 Nov 80-6 Aug 81</u>		12. REPORT DATE <u>11 6 August 1981</u>
		13. NUMBER OF PAGES
		15. SECURITY CLASS. (of this report) <u>Unclassified</u>
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) <u>Approved for public release, distribution unlimited.</u>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <u>Autonomy, DSCS III, autonomous spacecraft, autonomous satellite, fault protection, fault tolerance, on-board computing, autonomous satellite control, autonomous stationkeeping, autonomous health and welfare, redundancy management, autonomous navigation.</u>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <u>This volume assesses the ability of the next generation Defense System Communication Satellite (DSCS III) to be made autonomous, i.e., to operate successfully without ground intervention for up to six months. The existing DSCS III design will not operate autonomously for extended periods. This volume assesses the ability of the current DSCS III design to be upgraded for increased autonomy (1) without additional hardware and (2) with modest hardware additions and changes. Functional descriptions of options for increasing</u>		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

494150

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

BLOCK 20, ABSTRACT (continued):

DSCS III's autonomy are presented. Software-only modifications can make improvements in mission performance but cannot meet the six-month autonomy goal. Full autonomy hardware additions are necessary in three areas: (1) some additional autonomy for routine service functions, (2) addition of a fault tolerant autonomous spacecraft redundancy management function, and (3) addition of a fault tolerant autonomous navigation function. These functions can be added for relatively minor increases in mass and power, primarily providing additional computing capability and autonomous navigation sensors. On-board computing control complexity will be substantially increased. ↗

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

USAF REPORT SD-TR-81-87
JPL REPORT 7030-2

ASSESSMENT OF AUTONOMOUS OPTIONS
FOR THE DSCS III SATELLITE SYSTEM


VOLUME III: Options for Increasing
the Autonomy of the DSCS III Satellite

Accession For	
NTIS GNA&I	X
DTIC TAB	
Unannounced	
Justification	
By	
Distribution	
Availability Codes	
Dist	Availability for Special
A	


Prepared by
Donna L. S. Pivrotto, Assessment Team Leader
and
Michael Marcucci, Spacecraft Systems Engineer
Autonomous Spacecraft Project

6 August 1981

Approved:


Glenn E. Cunningham
Project Engineer
Autonomous Spacecraft Project

Approved:


David D. Evans
Manager
Autonomous Spacecraft Project

Prepared for
U.S. Air Force Systems Command
Headquarters, Space Division
Through an agreement with
National Aeronautics and Space Administration

by

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109

PREFACE

This document is the third of three volumes which make up the assessment of autonomy for the DSCS III satellite system. Volume I is an overview and summary of the assessment; review of it is recommended prior to reading subsequent volumes. Volume II is a functional description of the existing DSCS III satellite system and an assessment of its current autonomy. Volume III (this volume) presents options, at the functional level, for increasing the autonomy of DSCS III.

The DSCS III Assessment was performed by the Autonomous Spacecraft Project Team. Authorship of specific sections of the report by individual contributors is acknowledged in Volume II and in this volume. Unless otherwise noted, all contributors are JPL personnel.

CONTENTS

1.	INTRODUCTION	9
1.1	PURPOSE, FORMAT, AND USE OF THIS VOLUME	9
1.1.1	Purpose	9
1.1.2	Format and Use	9
1.2	ASSESSMENT FUNCTIONAL CLASSIFICATION	11
1.2.1	Levels of Autonomy	11
1.2.2	Importance	11
1.2.3	Difficulty of Implementation	11
1.3	SYSTEMS OPTIONS AND ISSUES OVERVIEW.	13
1.3.1	Goals vs Requirements.	13
1.3.2	Systems Options.	13
1.3.3	JPL Experience in On Board Computing vs Autonomy.	17
2.	AUTONOMOUS OPTIONS TO PROVIDE SPACECRAFT SERVICES	21
2.1	AUTONOMOUS OPTIONS TO PROVIDE POWER.	23
2.1.1	Collect Solar Power.	23
2.1.2	Condition Power.	23
2.1.3	Distribute Power	25
2.2	AUTONOMOUS OPTIONS TO PROVIDE ATTITUDE CONTROL	26
2.2.1	Stabilize Attitude	26
2.2.2	Maintain Stable Attitude During Normal Operations	28
2.2.3	Maintain Stable Attitude During Maneuvers.	28
2.2.4	Computer Resource Considerations for Autonomous Attitude Control	28
2.3	AUTONOMOUS THERMAL CONTROL OPTIONS	30
2.3.4	"As-Is" Options for On-Orbit Thermal Control . . .	30

2.4	AUTONOMOUS OPTIONS FOR S/C CONTROL AND MONITORING	32
2.4.1	Provide Telemetry Function.	32
2.4.2	Provide Command Function	34
2.5	AUTONOMOUS TIMING OPTIONS	35
2.6	AUTONOMOUS DIRECT PAYLOAD SERVICES OPTIONS.	36
2.7	AUTONOMOUS STATIONKEEPING OPTIONS	37
2.7.1	Sense S/C Position In Orbit	37
2.7.2	Direct/Analyze/Control Orbital Position of S/C.	37
2.7.3	Autonomous Maneuver Command Options	45
2.7.4	Autonomous Maneuver Options	45
2.7.5	Autonomous Stationkeeping Implementation Considerations	46
3.	MANAGE RESOURCES	49
3.1	AUTONOMOUS OPTIONS FOR POWER RESOURCE MANAGEMENT.	49
3.1.1	Manage Generated Power.	49
3.1.2	Manage Stored Energy	49
3.1.3	Manage Battery Life	51
3.2	AUTONOMOUS OPTIONS FOR PROPULSION RESOURCE MANAGEMENT	53
3.2.1	Manage Hydrazine Resources.	53
3.2.2	Manage Center of Mass	53
3.2.3	Manage Thruster Life	53
4.	AUTONOMY OPTIONS FOR INTEGRITY MAINTENANCE	55
4.1	AUTONOMOUS SPACECRAFT REDUNDANCY MANAGEMENT OPTIONS	55
4.1.1	A "Modest" Redundancy Management Design Architecture.	57
4.1.2	An Extensive Redundancy Management Design Architecture	71

4.2	AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF POWER FUNCTION.	94
4.2.1	Isolation/Protection from User Load Anomalies.	94
4.2.2	Reconfiguration Capability After Internal Faults	94
4.2.3	Maintain Battery Integrity.	97
4.2.4	Maintain Power Function Integrity	98
4.3	AUTONOMOUS OPTIONS TO MAINTAIN S/C ATTITUDE CONTROL FUNCTION	100
4.3.1	Perform Routine Health Checks and Maintenance	100
4.3.2	Configure S/C for External Events	100
4.3.3	Detect and Correct Device Failures.	102
4.3.4	Select ACS Modes and Modify Programs	102
4.3.5	Implementation Considerations for Autonomous Maintenance of the Attitude Control Function . . .	103
4.4	AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF THE S/C THERMAL CONTROL FUNCTION	119
4.5	AUTONOMOUS OPTIONS FOR MAINTAINING INTEGRITY OF S/C CONTROL AND MONITORING FUNCTION.	121
4.5.1	Maintain Telemetry Function.	121
4.5.2	Maintain Command Function	123
4.5.3	Options for Implementation of Autonomous	124
4.6	AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF THE S/C PROPULSION FUNCTION	138
4.6.1	Thruster Health	138
4.6.2	Propellant System Health	138
4.7	AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF STATIONKEEPING FUNCTION	140
4.7.1	Maintain Tracking Function	140

4.7.2	Maintain Autonomous Stationkeeping Function	142
5.	VALIDATION AND OPERATIONS ISSUES FOR AUTONOMOUS DSCS III SATELLITES.	144
5.1	PHILOSOPHY	144
5.1.1	Validation Philosophy.	144
5.1.2	Operations Philosophy.	145
5.2	VALIDATION/OPERATIONS METHODS.	147
5.2.1	Spacecraft State Simulation and Analysis	147
5.2.2	On-Orbit Verification of Autonomous Operations	147
5.2.3	Impacts of Spacecraft Autonomy on Payload Control and Utilization.	149
5.3	VALIDATION/OPERATIONS IMPACTS AND ISSUES FOR SOME AUTONOMOUS FUNCTIONS.	151
5.3.1	Issues in Validating Operation of the Autonomous Power Function.	151
5.3.2	Issues in Validating/Operating the Autonomous Attitude Control Functions.	152
5.3.3	Options for Operating an Autonomous Thermal Control Function	153
5.3.4	Ground Operations Considerations for an Autonomous Spacecraft Control and Monitoring Function.	153
5.3.5	Autonomous Navigation Considerations for Mission Operations	156
APPENDIX A	LEVELS OF AUTONOMY	159
APPENDIX B	SUMMARY OF JPL EXPERIENCE IN ON-BOARD COMPUTING VS. AUTONOMY FOR VIKING AND VOYAGER.	164

Figures

Figure	2-1. DSCS III Services Functional Hierarchy	22
Figure	2-2. Power Service Functional Hierarchy	24
Figure	2-3. Attitude Control Service Functional Hierarchy.	27
Figure	2-4. Thermal Control Service Functional Hierarchy	31
Figure	2-5. S/C Control and Monitoring Service Functional Hierarchy. .	33
Figure	2-6. Stationkeeping Service Functional Hierarchy	38
Figure	2-7. Autonomous Stationkeeping Functional Block Diagram	48
Figure	3-1. Resource Management Functional Hierarchy	50
Figure	4-1. Integrity Maintenance Functional Hierarchy	56
Figure	4-2. RMS Interface Block Diagram.	61
Figure	4-3. RMS Block Diagram.	62
Figure	4-4. RMM Block Diagram	63
Figure	4-5. I/O Module Block Diagram	65
Figure	4-6. System Design Architecture	75
Figure	4-7. RMS Block Diagram	77
Figure	4-8. RMM Block Diagram	78
Figure	4-9. I/O Module Block Diagram	80
Figure	4-10. DPU Block Diagram	82
Figure	4-11. Battery Hi-Temp Anomaly Contingency Events	91
Figure	4-12. S/C Power Integrity Maintenance Functional Hierarchy . . .	95
Figure	4-13. Attitude Control Integrity Maintenance Functional Hierarchy.	101
Figure	4-14. Voyager AACS Software Fault Monitor and Correction Algorithms	109
Figure	4-15. DSCS III ACS Software	
	(a). Executive Software Module	110
	(b). Attitude Controller (DCNTRL) Software Module Overall Block Diagram.	111
	(c). DCNTRL Block 11A Top Level Flow of Update Process	112

Figure 4-16.	Voyager Thruster Configuration	114
Figure 4-17.	Voyager Propulsion Subsystem	115
Figure 4-18.	Example DSCS III Thruster Fault Algorithm for Autonomous Reaction Wheel Uploading.	117
Figure 4-19.	Voyager TCAPIU Routine	118
Figure 4-20.	Thermal Control Integrity Maintenance Functional Hierarchy	120
Figure 4-21.	S/C Control and Monitoring Integrity Maintenance Functional Hierarchy	122
Figure 4-22.	DSCS RMS Functional Hierarchy	126
Figure 4-23.	DSCS RMS Functional Flow and Description	127
Figure 4-24.	DSCS TT&C MTU, RTU, and CD Functionally Redundant Blocks .	128
Figure 4-25.	Propulsion Integrity Maintenance Functional Hierarchy. . .	139
Figure 4-26.	Stationkeeping Integrity Maintenance Functional Hierarchy.	141
Figure B-1.	Viking Computer Command Subsystem (CCS).	165
Figure B-2.	Viking CCS Software Routine Structure	168
Figure B-3.	AACS Functional Block Diagram	178

Tables

Table 4-1.	RMS Implementation Characteristics.	70
Table 4-2.	RMS Implementation Characteristics	85
Table 4-3.	DPU Implementation Characteristics	86
Table 4-4.	DMS Performance Capability Summary	88
Table 4-5.	DMS Implementation Characteristics	90
Table 4-6.	DSCS III ACS and Voyager AACS Analogous Routines	107
Table 4-7.	DSCS RMS Fault Detection/Fault Correction Matrix for TT&C MTU, RTU, and CD.	129
Table 4-8.	DSCS RMS Functional Elements	130
Table B-1.	Viking Prime Mission Fault Routine Summary	169
Table B-2.	Viking Extended Prime Mission Fault Protection Summary . .	171
Table B-3.	Summary of Voyager CCS Fault Protection Algorithms	176
Table B-4.	Summary of Voyager AACS Fault Protection Algorithms . . .	180

SECTION 1

INTRODUCTION

1.1 PURPOSE, FORMAT, AND USE OF THIS VOLUME

1.1.1 Purpose

The purpose of this volume is to assess the ability of the existing DSCS III satellite design to be upgraded for increased autonomy.

- (1) Without additional hardware, and
- (2) With modest hardware additions and changes.

This volume contains a functional description of options for increasing the autonomy of the existing spacecraft. The options are presented in such a way that they could be implemented in increments. A series of changes to DSCS is planned by the Air Force to be implemented in future blocks. Although the options described in Volume III are not keyed to the specific block changes they could be implemented as part of a planned schedule of such changes.

1.1.2 Format and Use

The format of this volume is designed to describe each function in terms of the "sensing", "direction" and "action" functions as explained in Section 6.1.2 of Volume 1. While this format creates a lack of "flow" in the text, it has been selected for easy cross-reference with Volume II. The formats of Volumes II and III are identical, in that paragraphs devoted to each function have the same numbers in Volumes II and III. Therefore, the user can cross-reference between the current way the function is performed (Volume II), and the options for performing the function more autonomously (Volume III) by referring to the same paragraph numbers.

Volume I of this report should be read as background to Volume III. It presented a summary of possible ways to implement a phased program of additions to create a Level 5 spacecraft from the existing DSCS III. An alternative is to implement a new design wherein autonomy is included as a feature of a top-down system design. While the latter approach might be more expensive on a per step basis, its overall cost may be less than following the path of incremental additions.

The following sections of Volume III describe options for full autonomy and for a graduated series of partially autonomous steps to full autonomy. The options are referred to by paragraph number and include:

- 1.3.2.1 A Level 5 Autonomous Spacecraft (full autonomy).
- 1.3.2.2 The ACS Options (software changes and modest hardware add-on to existing subsystems).
- 1.3.2.3 The RMS Options (redundancy management subsystem add-on).
- 1.3.2.4 The Autonomous Stationkeeping Options (add-on).
- 1.3.2.5 The Phased Autonomy Option (add-on).
- 1.3.2.6 The Redesign Option (add-on).

A section (1.3.3) on JPL experience in the design and operation of autonomous spacecraft is included to provide a perspective on what can be accomplished with a relatively small amount of computer resources dedicated to autonomy, and to illustrate considerations in autonomy implementation.

1.2 ASSESSMENT FUNCTIONAL CLASSIFICATION

The DSCS III functions were classified in three ways: by level of autonomy, by importance, and by difficulty of implementation.

1.2.1 Levels of Autonomy

The levels specified in the Goals document (Reference 1) were applied to both the existing DSCS III and the autonomy options. These levels (from 0 to 10) are reproduced in Appendix A of this volume.

1.2.2 Importance

The primary requirement which drives the DSCS III autonomy is for the spacecraft to operate with reduced ground intervention. As stated in the Goals Document:

The autonomous spacecraft shall be capable of successfully performing the mission function for an extended period of time without ground support at a specified level of conflict. Specifically:

- (1) Autonomous spacecraft shall operate without performance degradation for up to 60 days from the last initialization update.
- (2) Autonomous spacecraft shall operate for up to 6 months from the last initialization update. They shall do so within acceptable performance degradation limits for mission-prioritized functions as defined by each mission.

These requirements were used as the basis for prioritization of autonomous operation as follows:

- (1) Category I: Functions which must be performed autonomously for the spacecraft to meet the 60 day/6 month requirement.
- (2) Category II: Functions which must be performed autonomously for lifetime protection (battery conditioning, etc.) or which, if performed autonomously, would increase the operability or operational flexibility of the spacecraft.
- (3) Category III: Functions not requiring autonomy.

1.2.3 Difficulty of Implementation

There are three modes by which the DSCS III satellite system can be made more autonomous: Software, Add-on, and Redesign. The first is to utilize the existing hardware capabilities of the system and make software

changes to increase autonomy. The attitude control subsystem includes a computer which is capable of being reprogrammed to increase the spacecraft autonomy (see Section 2.2 and 4.2). This mode will be referred to as the Software mode. It will produce the least expensive modification to DSCS III but is very restricted in its ability to add autonomous capability to the system. The Add-on mode adds hardware as well as software to the spacecraft but avoids making major design changes. The third mode, Redesign, allows consideration of redesigning the DSCS III system to increase its capabilities for autonomy. the Add-On and Redesign modes have gradations of difficulty. This assessment has classified hardware modifications as "modest" or "extensive".

For the purposes of the DSCS III Assessment task, "modest hardware" modifications may consist of the following:

- (1) New hardware introduced into the spacecraft system to perform autonomy functions, and/or
- (2) Modifications of hardware already existing in the spacecraft system.

In order to be classified as "modest," arbitrary constraints were defined:

- (1) The effects of added hardware would not allow the mass or power of the spacecraft to grow more than 5%, or the mass or power of the individual subsystem to grow more than 20%.
- (2) No more than 15% of the spacecraft system's electrical interfaces would be impacted.
- (3) If hardware is modified, the major function of that hardware would not be changed.
- (4) No more than 20% addition of piece parts would be allowed, and no more than 20% new electrical interfaces would be allowed.

Any changes with scope larger than a "modest" modification are referred to as "extensive."

1.3 SYSTEMS OPTIONS AND ISSUES OVERVIEW*

1.3.1 Goals vs Requirements

In order to meet the mission requirements for the required 60 days/6 months without ground intervention, the DSCS III spacecraft must have an overall autonomy level of about 5. The Goals document (Reference 1) lists goals for spacecraft at about Level 5. Some of the autonomy options described in this section will not meet all of these goals, but one option is presented which could meet all goals. Some goals have to do with the design rules for the autonomous spacecraft, e.g., "Autonomous spacecraft shall include reconfigurable software". To the extent that the options have been developed they meet these design-rule goals. In all options the basic mission requirements specified in Volume II would be met, but the incomplete autonomy options would still require ground intervention.

1.3.2 Systems Options

1.3.2.1 A Level 5 Autonomous Spacecraft. In order for a DSCS III Spacecraft to be capable of meeting its mission requirements for 60 days/6 months, additions to its capabilities are necessary in three areas:

- (1) Some additional autonomy for its service functions other than stationkeeping.
- (2) Addition of a fault tolerant, spacecraft redundancy management function.
- (3) Addition of a fault tolerant, autonomous navigation function.

The primary requirements for (1) and (2) are for the addition of computing capability and the associated sensor-to-computer data and control links. Addition of an autonomous navigation capability will require new sensors, as well.

Considerable autonomy can be added without requiring additional health/welfare sensors, but tradeoffs must be made between additional direct measurements of a spacecraft state condition and additional computing capacity to infer conditions from existing telemetry data. Addition of an autonomous navigation system will provide most of the functional autonomy needed to make all the service functions autonomous.

Computing tradeoffs will be necessary in the autonomous DSCS III design phase. The four major functions requiring computation are services (other than stationkeeping), resource management, redundancy management, and

*By D. J. Eisenman (G.E), C. P. Jones, E. C. Litty, E. Mettler, H. B. Phillips, and D. L. S. Pivrotto

stationkeeping. There are considerable overlaps between these functions. For example, hydrazine mass estimates are needed for planning maneuvers, for attitude control/operating mode selection, for propulsion system health maintenance, for center-of-mass management, and for hydrazine resource management.

An executive function will be required to manage these major computing functions. Distributed vs central data processing trades will be necessary in the autonomous DSCS III design process.

For the purposes of this assessment a Level 5 DSCS III spacecraft could be produced by the following:

- (1) Add the extensive Redundancy Management Subsystem (RMS) described in Section 4.1.2.,
- (2) Add the Autonomous Stationkeeping functions specified as Category I in Section 4.7, with fault tolerance included.
- (3) Upgrade the ACS microcomputer as described in Section 2.2, to handle the remaining Category I service functions, and
- (4) Add an executive computing system to manage the computing functions in (1), (2), and (3).
- (5) Some additional health and welfare sensors and data/control will probably also be required.

These steps would require at least a "modest" addition/modification of hardware, and will probably be "extensive", depending on the design approach. Trades will be needed to determine the best strategy for the DSCS III system to reach Level 5 autonomy. An example set of options for creating a Level 5 spacecraft was described in Volume I. Some aspects of these strategies are discussed here for context for Volume III.

1.3.2.2 The "ACS" Options. One option addressed was to make only software additions to the existing ACS microcomputer. The option of making only modest additions to the ACS computer to handle other autonomous functions such as redundancy management was also addressed. The capabilities of the ACS computer for expansion are discussed in Sections 2.2 and 4.3 and are summarized here.

RAM patching is the only way to add autonomous functions to the ACS without reprogramming the ROM. There are many limitations to using this method.

An appreciable amount of CPU time can be made available to perform autonomous functions as a background task (during non-RFN processing) without seriously changing the current ROM program timing or operating character. This can be done through RAM patch or ROM reprogramming. Doing so would increase the CPU/ROM average power usage by up to 60%.

The ROM space available for autonomous features is indicated to be less than 6% of the total ROM memory.

A significant portion of the ACS operating signals are available to the microcomputer either directly or indirectly. Inputs of ACS analog temperature measurements and a minority of digital inputs are not accessible without hardware modifications. The present microcomputer architecture does not lend itself to on-board management of CPU, ROM, and redundant ACS blocks by the ACS microcomputer itself.

To achieve some level of ACS redundancy management will require hardware modifications to provide access to the ACS DC relay matrix, EPDS DC relays or the TT&C command decoder.

The addition of I/O ports for redundancy control access is within the architectural expansion capability of the microcomputer. Addition of ports does have some clearly visible impacts to the system such as; power, weight, TT&C driver circuitry, ACS relay matrix, and program space. The feasibility of interfacing the I/O ports of the ACS to other subsystems is much less clear.

The addition of memory is within the architectural expansion capability of the microcomputer and has some clearly visible impacts to the system such as; power, weight, TT&C driver circuitry and ACS relay matrix.

Many ACS architectural changes (also TT&C and EPDS) would be required to perform ACS redundancy management for health and welfare maintenance. The input/output, and data storage and processing capabilities gained by these changes would also be achievable in a 'RMS' type subsystem (see 1.3.2.3, below). Trying to utilize the ACS microcomputer to perform spacecraft redundancy management and health and welfare maintenance would result in proliferation of input signal ports and interface lines from each subsystem. These inputs would be available to a 'RMS' type subsystem with far less hardware interface modification.

The ACS microcomputer with augmented I/O ports and memory for autonomous processing capability, used in concert with an 'RMS' type subsystem, would appear to be another means to perform both ACS and S/C redundancy management and health maintenance. The software-change-only option would have very limited capability to increase the autonomy of the entire DSCS III and could not meet the 60 days/6 months S/C autonomous operations requirement.

1.3.2.3 The Redundancy Management Subsystem (RMS) Option. Three options for adding a Redundancy Management Subsystem are discussed in Sections 4.1 and 4.5. The simplest system, that for the Spacecraft Control and Monitoring function only, can be easily extended to the spacecraft, and the option described in 4.1.1 appears to be the lowest level of RMS which should be considered. This option, which simply manages redundancy using straightforward inferences from existing telemetry data, could be of utility in reducing ground workloads. It could handle functions where fault detection and correction are both accomplished by simple trial and error switching of redundant elements. It could not deal with situations requiring inference from performance changes. It could probably not handle decisions requiring inputs from multiple sensors. However, it would be a "modest" hardware addition and is transparent to the existing system.

Addition of the "extensive" RMS described in Section 4.1.2 would create a much more flexible option. Because this RMS is modular and interacts with Distributed Processing Units (DPU's) devoted to major subsystems, its capabilities, (and its costs) depend on the degree to which it is implemented. At the lowest level it would be a "modest" hardware addition. In any case, addition of on board redundancy management without autonomous stationkeeping will not meet the primary autonomy goals. Long term, non-volatile, mass data storage will also be necessary for audit trails and program and parameter storage.

1.3.2.4 The Autonomous Stationkeeping Option. The autonomous stationkeeping functions described in Sections 2.7 and 4.7 are not as well developed as the RMS options. However, it is clear that extensive additions to the spacecraft will be required to meet the performance and autonomy goals.

Several options for implementing an Autonomous Stationkeeping System can be identified. The options vary depending upon the desired accuracy of the on-board orbit estimation process and the level of support to be provided to other elements of the spacecraft. Increased orbit determination accuracy requires both the use of more complex sensors (with accompanying mass, power and, possibly, structure implications) and increased computer capacity. Increasing support functions will require increasing only the computer capacity.

1.3.2.5 The Phased Autonomy Option vs the Redesign Option. The DSCS III program includes plans for extensive modification and/or redesign of the existing DSCS III. Reference 2 indicates that a total "production" of eleven DSCS III satellites is currently planned: two Block B, three Block C, three Block D and three Block E. Blocks B and C are not significantly different, in payload functions and basic spacecraft design, from Block A (the existing DSCS

III described in Volume II). A Super High Frequency (SHF) downlink is, however, added to the single channel transponder (SCT) for all blocks starting with Block B.

Block D incorporates significant product improvement which affect mass, power, and thermal characteristics. The propulsion system is planned to be changed from monopropellant to bipropellant to make a mass budget available for other product improvements. The power system capability also will be increased. Other than these, most of the planned improvements to Block D are in the payload. However, Reference 2 indicates that automatic redundancy switching is planned to be included to the extent feasible.

Block E is planned to retain all Block D DSCS III capability, and adds new features which will result in a major redesign of the spacecraft. Volume and mass may be increased as much as 50%. Added features are to include autonomous orbit determination and orbit adjust. The autonomous stationkeeping system is planned to provide 0.05 degrees or better attitude pointing accuracy, which is required for the planned payload antijamming features. A precision clock is planned to be added to support these requirements. Redesign of the attitude control electronics is planned as a part of a comprehensive system treatment of all on-board processing requirements. Extensive changes to DSCS III subsystems and structure will be necessary.

In addition, more participation by payload controllers in spacecraft control is under investigation. This will influence decisions on the extent to which spacecraft control functions, including telecommunications, should be made autonomous. Since redundancy management and autonomous stationkeeping are tentatively planned for the future DSCS III procurements starting in the mid 80's, it is apparent that the system options discussed in this section could be implemented in conjunction with other planned design changes. For example, the ACS computer capabilities could be expanded as early as Block C. The modest RMS could be incorporated in Block D, perhaps including distributed processing units. Autonomous stationkeeping could be added, as planned, in Block E, perhaps in conjunction with an expansion of the RMS.

1.3.2.6 The Redesign Option. Another approach is to design a fully autonomous system to be implemented at a point of planned major redesign (Block E). A mixed strategy of additions and redesign could also be employed. The relative costs of these approaches are not obvious. A phased addition program may be incrementally more cost effective but more expensive overall. Extensive analysis will be required in conjunction with the design phase to determine the relative costs of the possible approaches.

1.3.3 JPL Experience in On-Board Computing vs Autonomy

JPL's Mariner class spacecraft have always incorporated automatic features. On Voyager and Viking S/C fault protection algorithms were employed for critical spacecraft functions. The Galileo Project plans a somewhat more extensive fault protection effort. In order to provide a measure of the degree of fault protection which was available given the

Voyager and Viking S/C computer capacity the pertinent fault protection algorithms and their computing requirements are summarized in Appendix B. An overview is presented in this section.

1.3.3.1 Viking Fault Protection Experience

1.3.3.1.1 Computing Requirements. Viking orbiter included redundant CCS's with 4096 18-bit words each. The Viking prime mission fault protection routines required 818 words and occupied about 20% of one CCS. In the extended prime mission both CCS's were allowed to be used for fault protection. The extended prime mission fault protection algorithms used 1171 words and 14.3% of the combined CCS memory.

1.3.3.1.2 Design and Validation Strategy. During the Viking prime mission only one of the redundant CCS's was used for fault protection. Only critical spacecraft functions were covered and a large ground team was used. During the extended prime mission cost constraints resulted in a drastic manpower reduction (from 80 to 5, peak to end-of-mission). Therefore the extended mission experience is of most interest for an autonomous DSCS III. The fault protection system design goals were to:

- (1) Assure spacecraft safety (1st priority)
- (2) Minimize hands on operation.
- (3) Maximize science data return during remainder of mission.
- (4) Optimize use of expendables to extend spacecraft life.
- (5) Decrease peripheral ground support (tracking and computing).
- (6) Increase performance visibility.
- (7) Reduce costs.

The design strategy was to detect and correct failures at the subsystem level, whenever possible. Anomalies during the fault protection activities caused the spacecraft to revert to a "safe-hold" mode. The autonomous fault protection capabilities were added incrementally and revised throughout the extended mission.

The critical functions selected were (in order of priority) power maintenance, attitude control, gas preservation, spacecraft safing, preservation of communication, and protection against loss of data. Other fault protection functions managed the power resource, controlled spacecraft temperature, and managed CCS memory and the tape recorder.

1.3.3.2 Voyager Fault Protection Experience.

1.3.3.2.1 Computing Requirements. Voyager spacecraft has redundant CCS's with 4096, 18-bit words apiece. In addition the AACS computer is programmable and has redundant 4096, 18-bit words. CCS Fault protection routines used 1085 words (13% of the total memory). The AACS fault protection routines used about 794 words, or 20% of each AACS memory.

1.3.3.2.2 Design and Validation Strategy

1.3.3.2.2.1 Spacecraft System Fault Protection. The top level requirements on the Voyager fault protection design included one whose intent was to eliminate from the design all "single point failures" whose occurrence would result in the loss of more than half of the engineering data, or the data from more than one science instrument. Obviously, the requirement had to be waived when considering primary structure, the High Gain Antenna, the major elements of the Propulsion Module, and so on; but for electronic subassemblies the requirement was to be strictly adhered to. A second requirement dictated that whatever protection was to be provided had to be consistent with periods of unattended operation lasting from 24 hours during cruise to 10 days during superior conjunction. This requirement applied primarily to cruise-phase safing responses. During encounter periods, when round-the-clock tracking coverage was available, the long light time transmission delays became the significant design driver.

Finally, response priorities were established to direct the design. In order of decreasing priority, they were:

- (1) Spacecraft safety and commandability.
- (2) Preservation of spacecraft consumables.
- (3) Downlink telemetry visibility.
- (4) Ongoing sequence integrity.

1.3.3.2.2.2 Attitude Control System Fault Protection. The process for designing the AACS fault protection is described in Appendix B.

1.3.3.2.3 Conclusions From Voyager Experience. The Voyager fault protection system served the spacecraft well during flight. Since the system was designed to protect the spacecraft from critical faults and placed maximum reliance on the ground to restore normal operation, the ground manpower savings accomplished by the Viking Project were not evident. The system was capable of responding to anomalies caused by hardware failure, unexpected environmental conditions, and operator errors. During flight many of these responses were encountered with the spacecraft taking the expected actions. In several cases, however, the spacecraft responded unexpectedly due to decision thresholds being set too tight, or due to errors in data bases used

to design the spacecraft's activity. In other cases, the lack of good dynamic simulation of the spacecraft's attitude control subsystems caused a less than adequate test of the fault protection capability on the ground prior to launch and led to inappropriate in-flight responses. The ability of the system to accommodate changes to the fault protection routines because of their software nature was invaluable. The Voyager spacecraft has the capability to have its fault protection coverage greatly extended through access to the engineering telemetry in the same manner as was done on the Viking Project. However, because of competition for use of the memory space in the CCS by the science payload sequencing function, this extended capability has not been implemented.

In some cases, the lack of an "audit trail" function made ground analysis of the performance of the fault protection system difficult. The lack of precise data which caused entry into one of the fault routines, the order of occurrence of events, or the occurrence of a fault routine when the spacecraft was not being tracked led to more, not less, ground support being necessary. The need for improved filtering of data inputs and triggers of fault routines became evident. In some cases, noise transients (not actual problems) caused inadvertent entry into the routines. Since the need for fault protection varies with mission phase, it was determined that routines should likewise vary with mission phase to better complement or fulfill mission requirements by providing for variation in the priority, response time, and final condition (spacecraft state).

SECTION 2

AUTONOMOUS OPTIONS TO PROVIDE SPACECRAFT SERVICES

The current levels of autonomy of the Provide Services functions are described in Volume II, Section 2. Options for raising the autonomy levels of these functions are presented in this section. The options are intended to describe requirements and concepts for functions at about a Level 5 autonomy. Functions which are already autonomous to a sufficient level are discussed in Volume II. The current autonomy level and category of each function are stated in Volume III for reference.

Functions for which autonomous options are described are:

- (1) Power
- (2) Attitude Control
- (3) Thermal Control
- (4) Spacecraft Control and Monitoring
- (5) Timing
- (6) Direct Payload Support
- (7) Stationkeeping

Figure 2-1 displays these functions in a hierarchical form. Each function has its own, lower-level hierarchy displayed in the section which addresses that function. The numbers in boxes correspond to paragraph numbers.

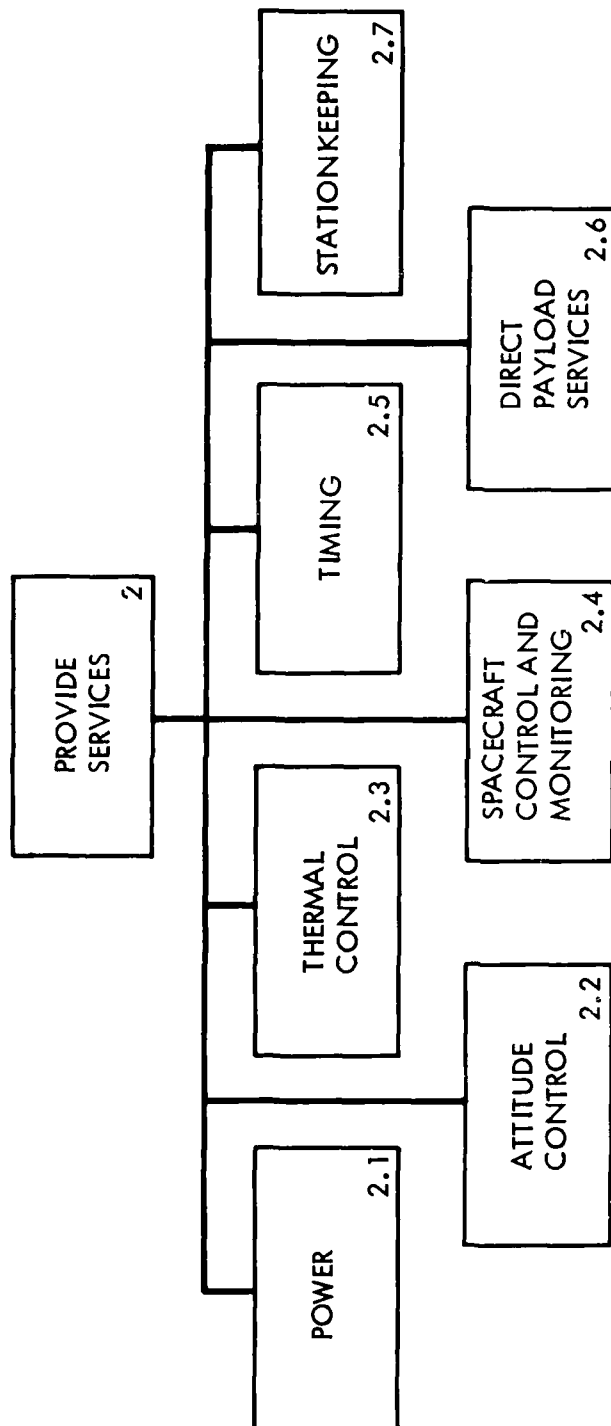


Figure 2-1. DSCS III Services Functional Hierarchy

2.1 AUTONOMOUS OPTIONS TO PROVIDE POWER*

Areas for which increased autonomy of the power function is required include:

- (1) Solar Array Orientation
- (2) Power Distribution

Other power functions are already autonomous to a sufficient level and will not be discussed here. Figure 2-2 shows the power function hierarchy.

2.1.1 Collect Solar Power - Level 4/Category I

2.1.1.1 (Post Separation) Orient Solar Array.

2.1.1.1 Control SA Articulation. Sufficiently autonomous.

2.1.1.1.2 Select SA Drive Mode. Solar array drive mode selection can be fully automated by addition of software to analyze/correlate requirements of mission phases, events, and timeline for issuance of SA position and rate commands to stepper motors A and/or B.

2.1.1.1.3 Select SA Drive Pot. The SA drive potentiometer selection can be automated by addition of fault analysis software and data channel selection logic.

2.1.1.2 Maintain SA Orientation. Sufficiently autonomous.

2.1.1.3 Correct Solar Array Misalignment. Sufficiently autonomous, based on ground calibration data taken pre-launch.

2.1.2 Condition Power

2.1.2.1 Regulate Main Bus Voltage. Control of main bus voltage is already sufficiently autonomous. (Level 5)

*By R. C. Detwiler, T. W. Koerner, and G. W. Wester.

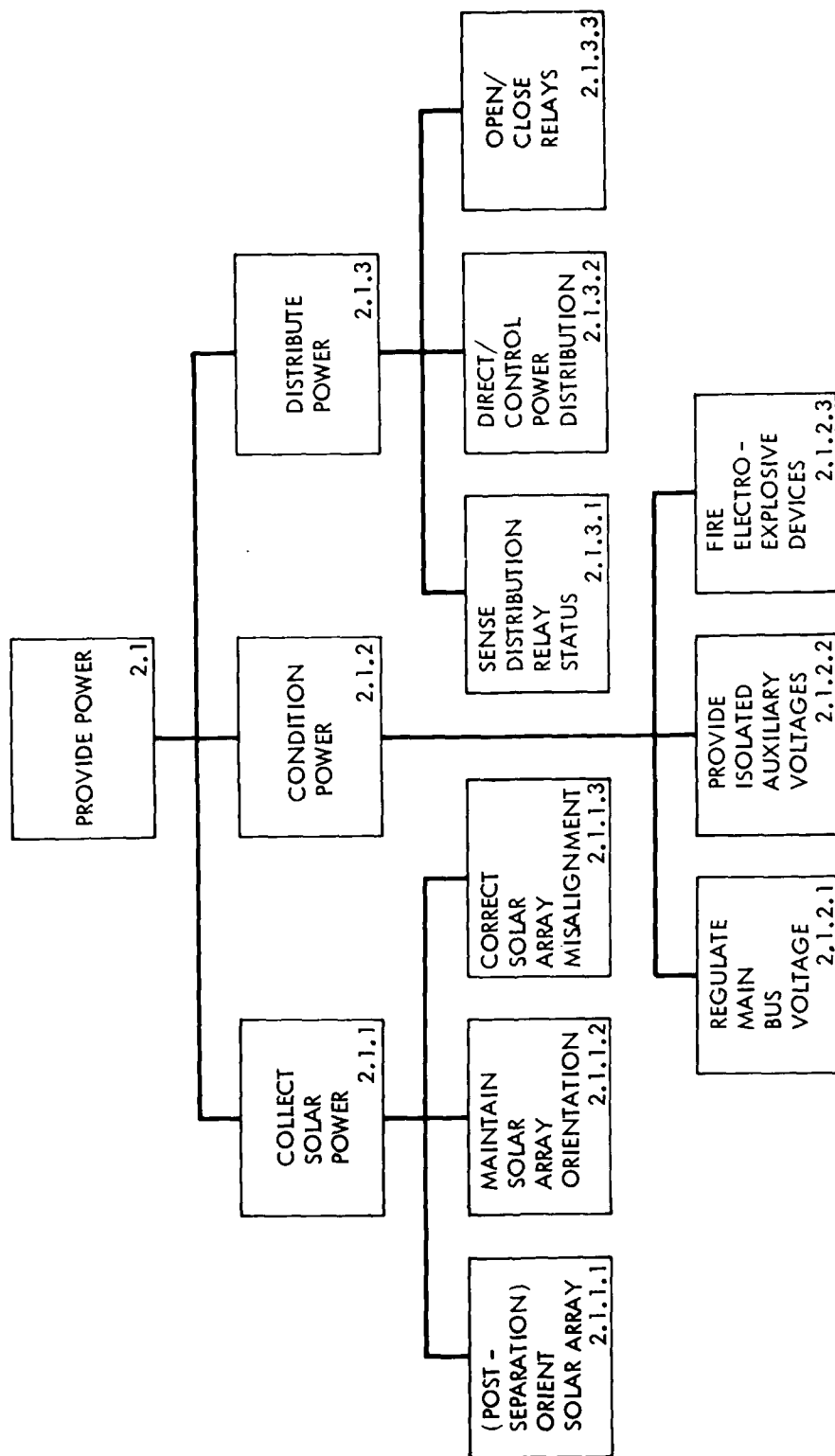


Figure 2-2. Power Service Functional Hierarchy

2.1.2.2 Provide Isolated Auxiliary Voltages - Level 5/Category I. Block redundant dc-dc converters provide the hardware redundancy needed in this area. Software to sense the necessity for switching is discussed under 4.2.2.3. Individual fault isolation is also covered in that section.

2.1.3 Distribute Power - Level 2/Category I

2.1.3.1 Sense Distribution Relay Status. Some combination of relay status signals and current/voltage signals for each load could simplify ground operations and also provide feedback for autonomous power distribution. Alternatively, with less hardware one might infer relay status from existing telemetry or bus current changes when a load is added or removed.

2.1.3.2 Direct/Control Power Distribution. For autonomous operation, power distribution (load management) is directed by various functional algorithms; e.g., stored energy management, battery life management, power function integrity. These algorithms indicate whether load management is needed and, if so, the quantitative change in either average or real-time load power. The power distribution function must then respond by considering alternative load configurations, selecting specific loads for load management, and scheduling load outages.

Additional capability required:

- (1) Load prioritization table
- (2) Load power for each operating mode
- (3) Timing function
- (4) Processing capability

2.1.3.3 Open/Close Relays. Relays could be controlled autonomously through existing control links.

2.2 AUTONOMOUS OPTIONS TO PROVIDE ATTITUDE CONTROL*

The DSCS III attitude control function is already highly autonomous in providing a stable platform for the payload. Only areas which are not currently autonomous will be addressed in this section. Paragraph numbers correspond to those of Section 2.2 of Volume II. Figure 2-3 is a hierarchy of the attitude control function.

Functions for which additional autonomy is needed include:

- (1) Post-launch earth acquisition,
- (2) Reference re-acquisition, and
- (3) Thruster selection.

Some options for utilizing/expanding the existing ACS computer for increased autonomy of the service functions are discussed in Paragraph 2.2.4.

2.2.1 Stabilize Attitude

2.2.1.1 Post Launch Acquire Sun and Reduce Tipoff Rates - Level 4/Category I.

2.2.1.2 Post Launch Acquire Earth - Level 3/Category I. The present ground enable can readily be deleted, and replaced by a stored command in the ACS computer which is issued after Auto sun acquisition is completed. It is the verification of the proper completion of sun acquisition that will require additional software (S/W) on-board to perform this analysis of parameters as a precursor to issuing the ENABLE. This development of analysis algorithms should be a moderate (not complex) task. Availability of computer resources in the present design will be the issue to resolve vs adding or redesigning the computer hardware.

2.2.1.3 Configure S/C for Normal Operations - Level 3/Category I. Reaction wheel selection and turn-on is enabled by ground command. This can be a stored autonomous function following completion of earth acquisition.

2.2.1.4 Reacquire References - Level 3/Category I. Re-acquisition of sun or earth can be automated in the same way as 2.2.1.2. Analysis S/W will be

*By E. P. Kan and E. Mettler

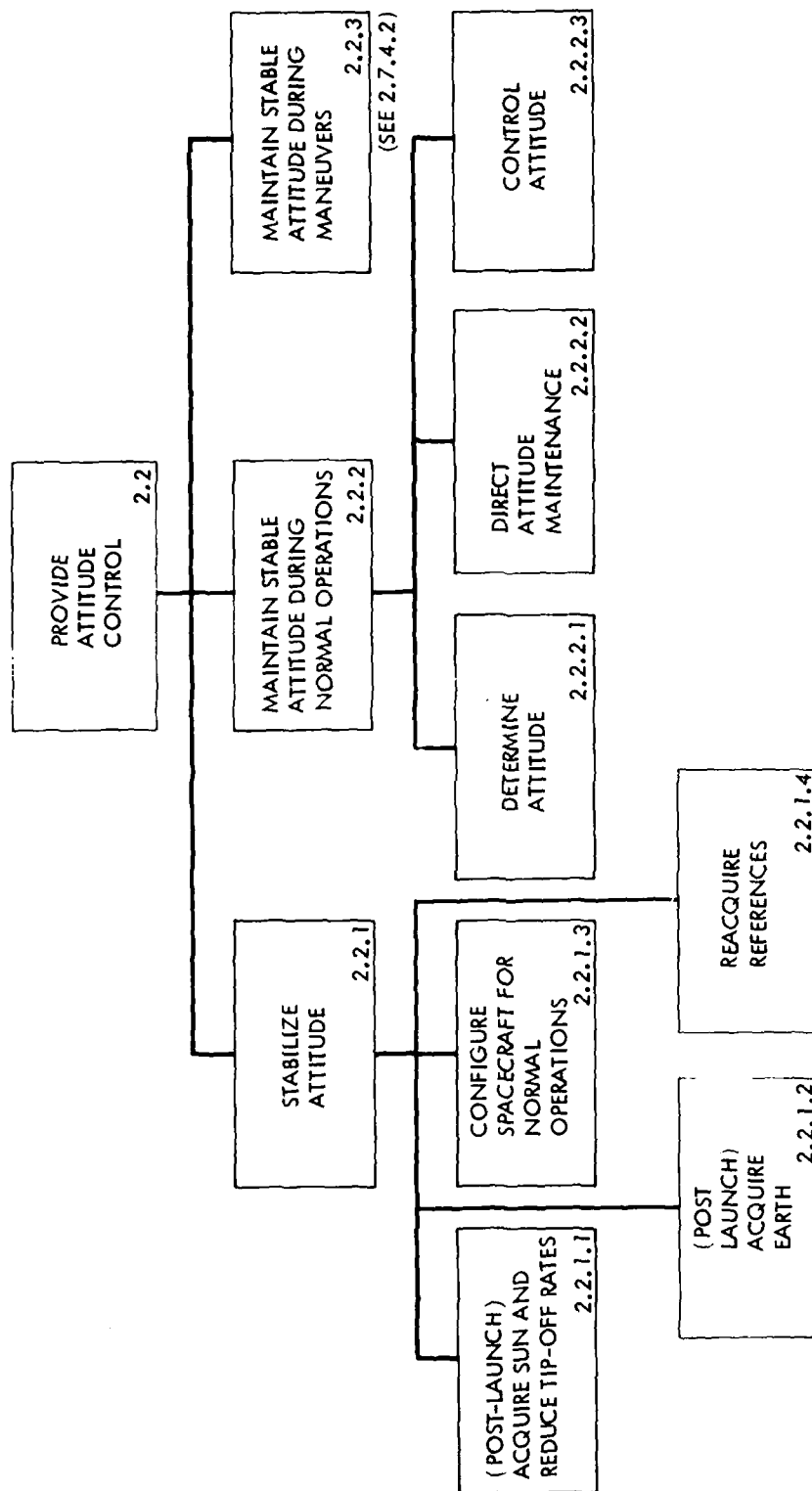


Figure 2-3. Attitude Control Service Functional Hierarchy

required for the precursor verifications. In particular, thruster performance estimates and propulsion resource and health status information would need to be output to attitude control. Selection of the rate gyro for yaw rate control in sun/earth reacquisition operations and sun sensor back-up can be automated by S/W added to do dynamic state analysis, and enable commanding of earth and sun search sequences.

2.2.2 Maintain Stable Attitude During Normal Operations

This function is sufficiently autonomous except for stationkeeping thruster selection, and change of attitude control and estimation parameters. Therefore, only Sections 2.2.2.2.4 and 2.2.2.2.6 are discussed here.

2.2.2.2.4 Manage Control Parameters - Level 0/Category I. This function can be automated by addition of S/W to analyze attitude state performance vs requirements and generate decisions for calling new values in algorithm variables based on control law relationships.

2.2.2.2.6 Select Thrusters for R/W Unloading - Level 2/Category II. In order for thrusters to be autonomously selected to offset E/W drift by reaction wheel unloading, an autonomous navigation function is required.

2.2.3 Maintain Stable Attitude During Maneuvers - see Section 2.7.4.

2.2.4 Computer Resource Considerations for Autonomous Attitude Control

2.2.4.1 Description of the Existing ACS Computer. The ACS computer is manufactured by GE and is based on a Digital Equipment Corporation (DEC) LS111 microcomputer system. A 1975 vintage Japanese space communication satellite used a similar design. JPL was directed to use the LS111 as the engineering model for the study. The ACS microcomputer system does resemble an LS111 type system but is tailored to function in its present manner for DSCS III. An ACS microcomputer architectural characterization is provided in Appendix B of Volume II.

2.2.4.2 Options for Performing Autonomous ACS Functions with Modest Design Changes.

2.2.4.2.1 RAM Patch. RAM patching could provide a 13% increase in available memory. Autonomous functions would vie for RAM space with other flight operational patching requirements. Because patching uses the redundant RAM while the prime RAM is also being used, access to it (an otherwise standby RAM) in the event of a prime RAM failure is not possible. RAM is vulnerable to

radiation effects and a power off/on occurrence. A detected nuclear event or a power off/on event will automatically disable all RAM patches. However, RAM patching does allow for flexible reprogramming and simple enable/disable of autonomous functions. Additional power, on the order of 5 watts, is required to operate the redundant RAM for patching.

2.2.4.2.2 CPU/ROM. During the busiest CPU execution cycle approximately 5% (54 milliseconds) of the CPU time is not consumed. Programs to perform autonomous functions could execute to completion each CPU execution cycle or in parts through successive CPU execution cycles. However, these programs would vie for the CPU time with other flight operational requirements to perform RAM patching, i.e., some CPU time must be reserved for flight contingencies.

Alternatively, the 500 millisecond (50% of the CPU execution cycle) normally denoted to the BFN functions could be used for autonomous functions in the absence of a ground commanded request for BFN program execution.

As outlined in Appendix B of Volume II, the CPU and ROM are power strobed which indicates a considerable concern for power usage. Increasing the CPU/ROM execution time by 5% to 50% would increase the average power usage from the current 6 watts to between 8 and 14 watts.

The ROM space available to accept autonomous functions is not likely to be available. However, LS11 type ROM comes in increments of 512 words and each increment requires about 1.5 watts of power. The fact that 8K of ROM is now implemented could indicate that available ROM space of between 0 and 511 words, as a margin, may exist.

2.2.4.2.2.1 Input. All ACS bi-level indications are not directly presented to the microcomputer. Most (85%) are indirectly presented through the discrete command latching relay matrix. Most (75%) ACS analog signals are not presented to the microcomputer. Analog signals which are presented are digitized through a sequential signal buffer, MUX, A/D converter arrangement which make up the sensor port. The analog signals not presented to the microcomputer are temperature measurements.

2.2.4.2.2.2 Output. The microcomputer has no control access to manage redundant ACS blocks. Management of redundant ACS blocks, which include redundant microcomputer blocks, must be done via the ground segment. The microcomputer can send and receive data from those redundant blocks which are, or could be, always active; RW, BFN and SAD ports. It can alter its own processing by self-generating modifications to RAM parameters (message commands). The microcomputer can also modify its own processing of discrete command (DC) status indicators but only for those which do not require selection of redundant hardware, e.g., discrete commands which select the 2 second or 16 second outer control loop processing.

2.3

AUTONOMOUS THERMAL CONTROL OPTIONS*

Figure 2-4 shows a hierarchy of the thermal control service functions. The DSCS III thermal control subsystem autonomous capability can be introduced with a minimum of impact on thermal control subsystem.

Basically, the thermal control subsystem is designed for autonomous operation. The design consists of a passive thermal control system that controls the energy balance to the space environment, with an active thermal control system consisting of electrical heaters to adjust for variation in internal dissipation and from external sources.

To make the thermal control subsystem completely autonomous, the control heater system must be commanded on at all times. This will allow the maintenance of the spacecraft and its components within operational limits for an operational condition. The survival heater system is always on, thus this system requires no changes.

Control Heater Circuits once commanded on must be capable of being deactivated if the spacecraft puts itself into a survival condition. Prelaunch, launch and orbit transfer phase thermal control are sufficiently autonomous (2.3.1 through 2.3.3) and only Section 2.3.4 will be discussed here. The following are options/classifications of changes for Level 5 autonomous operation of the on-orbit DSCS III thermal control functions.

2.3.4

Options for On-Orbit Thermal Control

2.3.4.1 Hardware Additions for Autonomous Thermal Control - Level 3. This option will require additional computing capacity. It would include development of hardware for computer control of the thermal control subsystem. All heater control logic and temperature sensors would be routed to this computer. This computer would be used for thermal control of the spacecraft (redesign-extensive hardware change.)

2.3.4.2 Autonomous Operation of the Catalyst Bed Heaters - Level 1/Category I. The catalyst bed heaters must be activated 100 minutes prior to any thruster operation. A command must be in the logic to turn on the catalyst bed heaters if the propulsion system is activated, with a 100 minute time delay prior to thruster firing (minor change). This option can be accomplished with software changes to the existing ACS computer.

*By R. N. Miyake and J. A. Plamondon

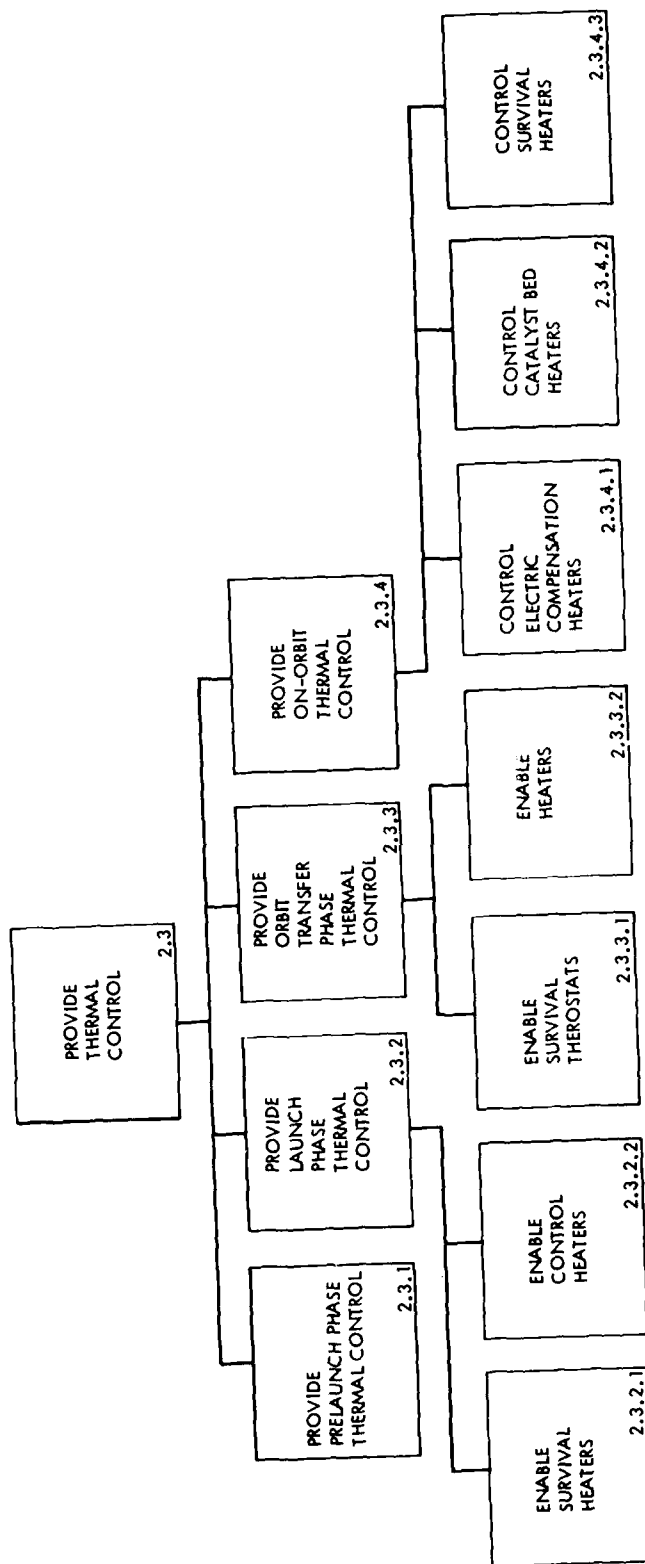


Figure 2-4. Thermal Control Service Functional Hierarchy

2.4

AUTONOMOUS OPTIONS FOR S/C CONTROL AND MONITORING*

The basic S/C functions of receiving commands and transmitting telemetry are essentially autonomous once these functions have been operationally established by ground based actions. There essentially are no options for additional autonomy for the S/C functions, except for the one below. Functions which must continue to be performed by the ground (e.g., ground telemetry processing, ground command operation) are not addressed as candidates for autonomy. Figure 2-5 shows a functional hierarchy of S/C control and monitoring.

2.4.1

Provide Telemetry Function

The S/C telemetry service functions are autonomous (see Volume II, Sections 4.1.1, 4.1.2, 4.1.3). A possible exception and an autonomous option are presented below. The telemetry reception, processing and distribution are ground functions and are not being addressed as candidates for autonomy. Information acquisition and telemetry generation are sufficiently autonomous, and Sections 2.4.1.1 and 2.4.1.2 are not discussed here.

2.4.1.3

Autonomous Send Telemetry Option - Level 3/Category II. The basic sending (or transmitting) of telemetry information by the S/C is autonomous once activated by ground command. The S-Band telemetry transmitter is normally off and activated automatically by establishing an uplink command signal. The S-Band also can be turned on or off by direct ground control and can be turned on by the "loss of earth presence," 80 minute battery timer or the launch initiation timer. Both of the X-Band telemetry transmitters (Beacons A and B) are normally "on". They can be turned on or off by ground command and can be turned off by the "loss of earth" or the 80 minute timer. Both the S and X-Band transmitters could be designed to be turned off by some of the possible power or temperature autonomy options.

If it is assumed that telemetry is required to be returned independent of uplink action then some autonomy would be needed to get the transmitters back on the air if they were shut down by S/C fail-safe actions. An on/off sequencer could be used to provide a simple form of autonomy. One approach would be to have the S and X-Band telemetry function powered continuously. They would only be turned off by S/C power or temperature fail-safe routines. The on/off sequencer would be programmable by ground command to turn the X and S-Band on or off by command variable time durations and command variable times. The "sequence" would be established by ground command and updated by the ground as required by mission operations.

*By S. O. Burks

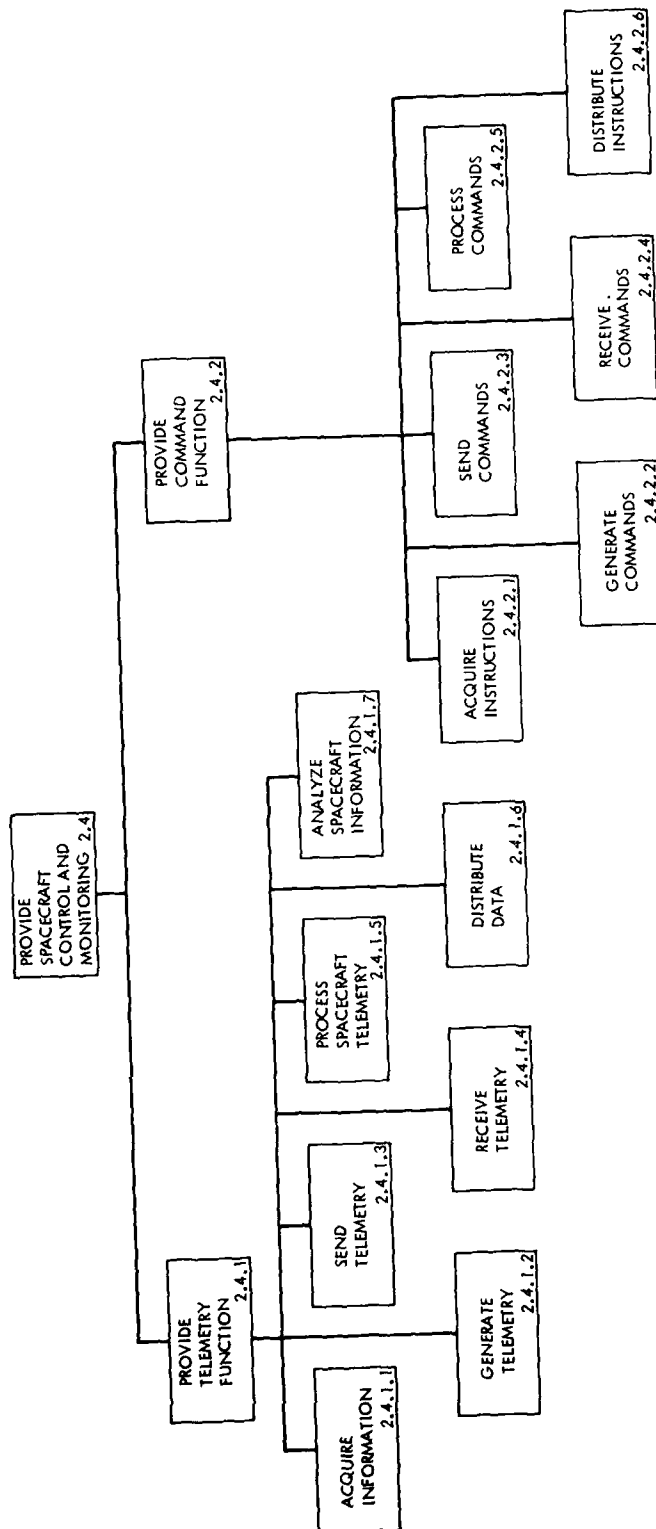


Figure 2-5. S/C Control and Monitoring Service Functional Hierarchy

For some of the options provided in Section 4.5.3 for telemetry maintenance options, the on/off sequencer would be required to periodically provide a "self-test" of the telemetry function.

This type of variable duration and time sequencer is fairly simple and could probably be added to the command decoder or a "Redundancy Management Subsystem" for less than 0.5 Kg mass and 0.5 watts power increase. The technology for this function is readily available and testing is relatively easy.

2.4.2 Provide Command Function

The functions of acquiring instructions, generating commands, and sending commands, (Sections 2.4.2.1 through 2.4.2.3 of Volume II) are ground functions and are not addressed here. The functions of command reception, command processing, and command issuance (Sections 2.4.2.4 through 2.4.2.6 of Volume II) are completely autonomous.

2.5

AUTONOMOUS TIMING OPTIONS

The current timing function is autonomous. Additional timing capability may need to be provided for autonomous functions (see especially Section 2.7 Stationkeeping).

2.6

AUTONOMOUS DIRECT PAYLOAD SERVICES OPTIONS

Reorientation and reconfiguration of the payload antennas are payload control functions and are outside the scope of the DSCS III assessment activity.

2.7 AUTONOMOUS STATIONKEEPING OPTIONS*

Since stationkeeping (navigation) is totally a ground function at present, all of the sensing and direction/control functions for autonomous stationkeeping will have to be added to the spacecraft. This section discusses the functions to be made autonomous and their requirements. Some options for implementing autonomous stationkeeping are described briefly in Volume I and Section 2.7.5 of this volume for purposes of assessing potential impacts on the current DSCS III design. Actual design of options for autonomous stationkeeping will be addressed in the Autonomous DSCS III design task. Figure 2-6 is a hierarchy of the stationkeeping function.

2.7.1 Sense S/C Position In Orbit - Category I.

2.7.1.1 & 2.7.1.2 Track S/C. Ground tracking of the S/C may be maintained as an option.

2.7.1.3 Sense S/C Position On-Board S/C - Level 0/Category I. Requires navigation sensors and appropriate interfaces to be on board the spacecraft for autonomous navigation.

Navigation sensors must be provided to supply measurements of sufficient accuracy for navigation. The number of sensors, types of measurements, and frequency of observations are major navigation subsystem design trade-offs. Use of data from existing or upgraded attitude determination and control sensors implies a new interface and design change to the current ACS. This may not be as effective as providing a separate set of sensors used only for navigation.

2.7.2 Direct/Analyze/Control Orbital Position of S/C - Level 0/Category I

A subset of the DSCS III Ground Navigation functions must be moved on-board along with sufficient computational capability to support them. Some current ground functions will be unnecessary for autonomous Navigation and will remain ground based analysis/validation support functions. A significant number of functions may be moved on-board to support autonomous maintenance of the Navigation or other subsystems. These, however, add significantly to the cost and complexity of new interfaces, in addition to increasing the level of autonomy. Finally, a subset of direction/controlling functions must remain ground-initiated by their nature -i.e., station repositioning.

A subset of current ground navigation functions must be supplied on board. The following functions are required on board to some degree for an autonomous subsystem.

*By J. B. Jones and P. R. Turner

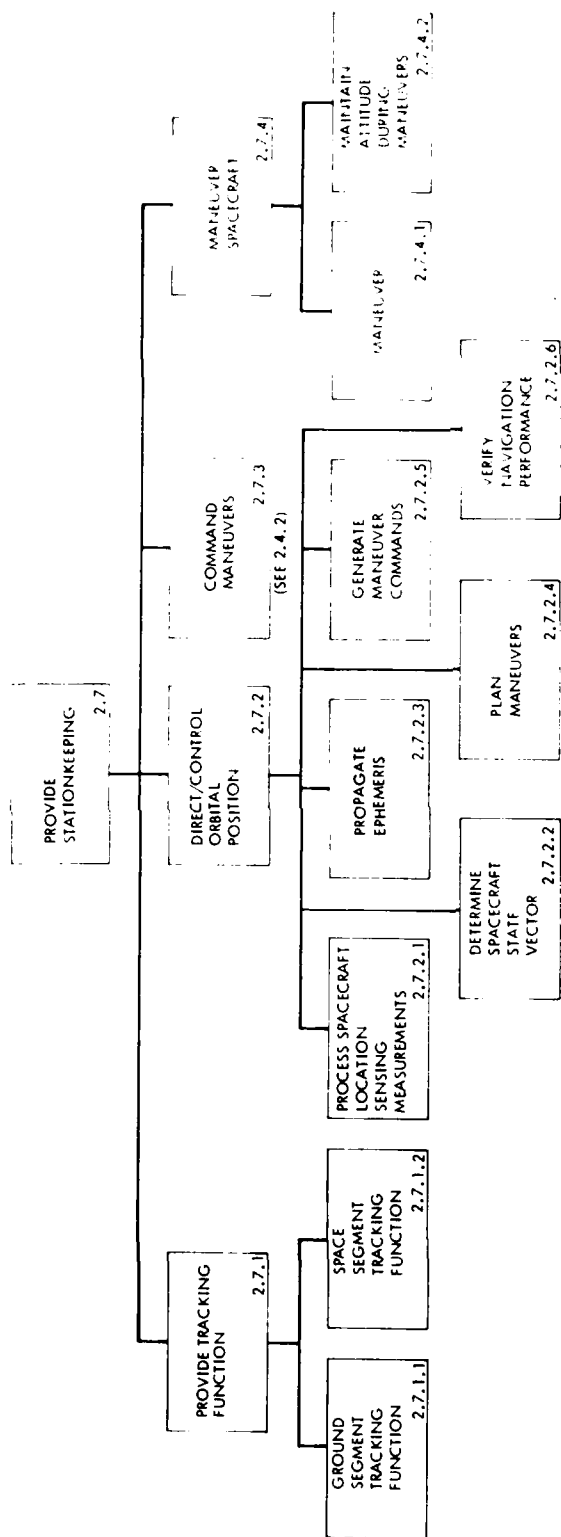


Figure 2-6. Stationkeeping Service Functional Hierarchy

2.7.2.1 Sensor Measurement Processing - Category I. Sensor measurement may consist of raw data from a conventionally designed sensor or "extracted" measurements from a sensor including a level of pre-processing to pick out desired measurement quantities in the raw sensor output. The first interface requires a navigation subsystem function of extracting the desired measurements. The second requires the subsystem to edit extracted measurements based upon data suitability criteria and navigation data processing strategy algorithm.

A major subsystem design trade-off is also whether measurements will be made and processed continuously or at intervals specified by a navigation control algorithm. The first option implies a continuous navigation process with redundant data while the second allows sensing and orbit determination to be a longer term periodic function occurring no more often than needed to meet accuracy requirements.

2.7.2.1.1 Data Acquisition Scheduling - Category I. This function must be available on-board. The complexity of implementation will depend upon whether the selected Navigation configuration utilizes continuous or periodic measurements and upon the nature of the measurement types. This is a candidate function for a Navigation subsystem Executive control function.

2.7.2.1.2 Measurement Data Request - Category I. A subset of 2.7.2.1.1.

2.7.2.1.3 Process Sensor Measurements - Category I.

2.7.2.1.3.1 Propulsion Subsystem Telemetry Processing - Category I. Possible for several designs with different Navigation/Propulsion subsystem interfaces. Currently, telemetry data for fuel tank quantities and configuration and thruster status and thruster performance estimates are required. Redesign for on-board application will require dividing detailed functions among Navigation and Propulsion.

This function is a design trade-off among navigation, attitude control, and propulsion. It is tied in directly with functions 2.7.2.5.1.1 and 2.7.2.5.1.2

2.7.2.1.3.2 Navigation Sensor Measurement Processing -Category I. Some Navigation specific editing and validation of sensor data will be required.

2.7.2.1.4 Update Data Base with Telemetry/Sensor Data - Category I. In addition to a requirement for normal Navigation data processing, this can support spacecraft ASM features and the audit trail requirement for ground validation of autonomous operation.

2.7.2.2 Spacecraft State Determination - Category I. Autonomous state determination may be performed continuously via a filter process at a rate to be determined, or less frequently via a batch process. The trade-off will be a major design issue.

Environmental models will have to be provided for this function as well as for the ephemeris propagation function. The degree and complexity of models required will depend to some degree upon the measurement types and orbit determination strategy that is selected.

Output of the determined state at an epoch must be saved and supplied to an outside user in the ground system and maintained on-board for navigation subsystem as well as fulfillment of audit trail requirements.

2.7.2.2.1 Maintain Data Editing Controls and Model Information - Category I. Some form of executive control over software data inputs will be required for adaptive navigation strategy in autonomous operation. A simplified version with constants modifiable from the ground at long intervals would meet lower level autonomy goals.

2.7.2.2.2 Provide Environmental Models - Category I. Earth gravity Luni-Solar Perturbation, velocity change, and other numerical models must be available to support this function (2.7.2.2) and others.

2.7.2.2.3 Determine State at an Epoch - Category I. The exact method to be used is greatly dependent upon selection of measurement types and overall Navigation Strategy. This is a major design trade-off issue.

2.7.2.2.4 Update Navigation Software Data Base - Category I. The newly determined state and information for a Validation Audit of Subsystem performance must be maintained at intervals dependent upon the overall Navigation strategy.

2.7.2.3 Ephemeris Propagation - Category I. The type and degree of accuracy required of environment models will depend upon the accuracy of the input state vector provided by orbit determination and the length of time over which it must be propagated to a specific accuracy. The propagation time and accuracy issues will be dependent upon how often the stationkeeping limits must be checked and how far ahead the maneuver planning algorithms must project the result of a maneuver.

North-South and East-West station limits violations must be checked at least at some minimum frequency and far enough ahead to allow time for maneuver planning and execution if a violation is predicted for the span.

Modeling of momentum dumping may be required, but the cost of autonomous thruster bank switching must be assessed to determine whether a canned procedure (such as alternating banks each day) would be as effective as computing the optimum switching point. This should be based on a propellant usage criteria, as the number of maneuvers alone is not as important for an autonomous system as for the current ground system.

Prediction of navigation sensor interference due to eclipses and occultations as well as time and location of nodes, etc., will be required to support both ASM goals and maneuver planning. This event data must be maintained on-board for use until an update is required or computed each time it is required.

2.7.2.3.1 Model S/C and Physical Environment - Category I. Required for dynamic modeling to propagate a state vector.

2.7.2.3.2 Calculate Station Limits Violations - Category I. Station violations must be predicted to schedule stationkeeping maneuvers and possibly to schedule execution of maneuver planning functions.

2.7.2.3.2.1 North-South Violations on Inclination Bounds - Category I. This is a critical feature due to its impact on propellant usage being considerably greater than for East-West longitude violations.

2.7.2.3.2.2 East-West Longitude Violation - Category I. This feature is less fuel-sensitive than inclination control. However, a trade-off needs to be made with respect to the requirement to model momentum dump thrust V's and whether or not it is cost effective to autonomously control thruster bank switching.

2.7.2.3.3 Predict Orbit Related Events - Category I. Some event prediction will be necessary to support Navigation subsystem control. Other predicts of celestial and orbit-related items may be useful to ASM or other subsystems.

2.7.2.3.3.1 Predict Antenna Pointing Requirements - Category III. The payload antenna pointing could be updated several times a day to compensate for diurnal orbital motion. This could be used to increase end-of-life performance or to save propellant by opening up stationkeeping limits.

The current antenna pointing calculations are for controlling ground-based S-Band TT&C antennas. An autonomous on-board navigation capability would probably not do this, but on-board state knowledge would allow the payload antenna pointing to be adaptively modified to respond to diurnal satellite orbital motion. This could allow a relaxation of orbit

control requirements on inclination and longitude stationkeeping. This would save fuel and increase spacecraft reliability by requiring fewer maneuvers to be performed over a given span of satellite life. An interface with the ACS would be required and greater memory requirements might result for the ACS computer.

2.7.2.3.3.2 Predict Sun/Moon Interference with Sensors - Category I. This will support ASM validation of Navigation sensor behavior and may provide a service for Attitude Control

2.7.2.3.3.3 Predict Eclipse/Occultation Events - Category I. Same purpose as 2.7.2.3.3 for Navigation. Can also provide support for attitude control and power subsystems. Note that maneuvers cannot be scheduled to occur during eclipses.

2.7.2.3.3.4 Provide State Vector to Users - Category I. The state vector could be computed as a by-product of the onboard orbit determination process. The frequency of update would depend on the accuracy of the on-board system and the stationkeeping maneuver schedule. The data would be available on the downlink to the users and could be provided in a form convenient for user processing.

2.7.2.3.4 Supply Event Data to Subsystem - Category II or III. Complexity of interfaces and design requirements of other subsystems are a primary Spacecraft System level tradeoff.

Navigation related data on eclipse/occultation periods, spacecraft orbit state, maneuver occurrences, lunar/solar ephemeris, and related issue can be supplied to external subsystems if their autonomous design requirements call for such data. Specific event data interfaces might be Category III due to the cost or complexity of redesign or implementation.

2.7.2.4 Maneuver Planning - Level 0 - Category I. Stationkeeping maneuvers must be scheduled and sized for velocity requirements. Inclination, longitude, and eccentricity must be controlled by the resulting maneuvers and spacecraft related constraints must be considered in the planning process. Hopefully, the constraint on maneuvering within restricted sun regions can be removed. If an alternate control scheme, such as Polaris sensing, is utilized by the ACS, it could also remove the restriction on not maneuvering in eclipse period.

Data base values for current constraints and any vehicle related variables must be updated before maneuver computation, and results of the maneuver planning process must be maintained for maneuver command generation and for maintaining an audit trail for validation. Autonomous maintenance concerns can be enhanced by propagating the predicted post-maneuver orbit and by instituting checks on the magnitude, direction, and time of the planned maneuvers.

2.7.2.4.1 Station Acquisition and Repositioning - Category III. This would require an increase in cost/effort above the normal stationkeeping function.

These functions require ground initiation and may also require changes to the on-board data base. Significant new control logic may be required to implement the drift cycle to acquire a station from either a drift orbit or an old station. This function can be provided on-board, and may provide a useful update capability if the navigation computer is sized with growth capability beyond the Category I requirements.

2.7.2.4.2 North-South Inclination Stationkeeping - Category I. Time and ΔV requirement to maintain inclination within bounds is required.

2.7.2.4.3 East/West Longitude Stationkeeping - Category I. Time and ΔV requirements computation is required.

2.7.2.5 Command Parameter Generation - Category I. This must be provided on-board, but there is a major trade-off of responsibility to be made between the amount of work required of navigation and that of propulsion and attitude control. The simplest possible implementation for navigation would be to limit its responsibility to supplying the effective time, magnitude, direction, and type of a velocity maneuver to an external interface. The most complex would be for navigation to maintain a propulsion subsystem status model, calculate the maneuver start time and duration, select thrusters and tank configuration for center of gravity maintenance, and supply an integrated command sequence to an external interface.

2.7.2.5.1 Maneuver Command Generation - Category II. A trade-off must be made between responsibilities of navigation, attitude control, and propulsion subsystems. This function must be autonomous but is not necessarily a navigation responsibility.

This function encompasses propulsion data base maintenance, propulsion subsystem performance modeling, and generation of an integrated sequence of ACS and propulsion subsystem commands to accomplish ΔV maneuvers. These functions require a major functional design trade-off between navigation, propulsion, and attitude control.

2.7.2.5.1.1 Propulsion Model Maintenance - Category I. This is broken down into propellant quantity assessment, center of gravity control, and tank/thruster valve operability and status. All these functions must be provided autonomously, but there is a major design trade-off between allocating the responsibility to navigation or propulsion.

2.7.2.5.1.2 Model Propulsion Performance for V - Category I. The propulsion subsystem performance in its status for the maneuver must be modeled to determine burn initiation time, burn duration, and fuel usage. This is another major trade-off decision between propulsion and navigation.

2.7.2.5.1.3 Assemble Total Command Sequence for V - Category I. Propulsion and attitude control system configuration, thruster selection, duration, and initiation commands must be assembled into an integrated sequence. This is again not necessarily the responsibility of the navigation system.

2.7.2.5.1.4 Supply Command Sequence to Sequencer - Category I. Ditto above comments.

2.7.2.6 Verify Navigation Performance -Category II. This function would provide support for autonomous maintenance fault detection and provide for an autonomous adaptive navigation strategy.

2.7.2.6.1 Maneuver Accuracy Assessment - Category II. Some degree of comparison of post maneuver state with predicted pre and post-maneuver states would allow independent assessment of navigation and propulsion subsystem performance to support subsystem and overall autonomous maintenance goals. More detailed support for thruster calibration, fuel usage monitoring, navigation strategy modification, and support of other subsystems would be Category III.

This function would "close the loop" on the navigation process by determining how closely the executed maneuvers were matching the commanded maneuvers. The check would support autonomous maintenance detection of loss of thruster performance or command generation errors. It would also allow for an adaptive maneuver generation capability by accomplishing thruster calibration with changes in propulsion performance over the vehicle lifetime.

2.7.2.6.2 Ephemeris Accuracy Assessment - Category II. Could be provided to support an adaptive navigation strategy. This function might enhance adaptive navigation strategy or act as an autonomous maintenance performance check on the navigation process. Basically the function would periodically compare the results of orbit determinations with a previous orbit determination propagated to the new epoch. The result of the comparison might reveal a need for a change in orbit determination or overall navigation strategy. It could also reveal a malfunction or numerical difficulty in either the orbit determination or ephemeris propagation function.

2.7.2.6.3 Assess Propellant Status Effect on Navigation Strategy - Category II. End-of-life propellant conditions would modify orbit determination frequency and stationkeeping limits and strategy.

This would fit in with an adaptive navigation strategy to support both end-of-life propulsion system performance and failures of the tankage or plumbing causing loss of some capabilities. An example might be to terminate inclination stationkeeping when propellant available falls below a pre-determined level.

2.7.3 Command Maneuvers - Level 0/Category I

Maneuver commands could be issued through the ACS computer or Command Decoder, or through direct links to the propulsion function -see discussion in Section 2.7.2.5 of Volume II.

2.7.4 Maneuver Spacecraft - Category I

The direct action of commanding thruster firing by the Navigation Subsystem would not be a practical design due to ACS control of thrusters for attitude control, at present. Some sort of command interface between Navigation and the ACS/Propulsion Subsystem should be furnished rather than providing an action function for Navigation. Direct action by the navigation subsystem is a design trade issue for generic autonomous design. For DSCS III it is probably more appropriately Category II. Prospects for direct action include thruster selection, sensor management, and ASM support.

Thruster commanding for DSCS III is currently under control of the ACS and there is no valid reason to change that responsibility. An autonomous navigation function could be utilized to act in controlling its own sensor data acquisition, adaptability to eclipse/occultation conditions, and fault identification/redundancy switching.

2.7.4.1 Select Thrusters - Level 0/Category I.

2.7.4.2 Maintain Attitude During Maneuvers - Level 3/Category I. It is possible to automate Initial Orbit Adjust, with thrusters and attitude control loop selections, contingent on having the addition of an autonomous Navigation subsystem to the S/C. This is a major capability improvement requiring a dedicated NAV processor and appropriate interfaces with ACS.

Velocity adjust for stationkeeping/station change has the same contingent factor.

Automating selection of S/C earth pointing control loop options during unbalanced velocity thrusting (thrust vector not coincident with S/C mass center) is also contingent on adding an autonomous navigation system.

2.7.5

Autonomous Stationkeeping Implementation Considerations

The dominant requirements of an autonomous stationkeeping system for the DSCS III spacecraft are to maintain the longitude station within $\pm 0.1^\circ$ and to supply the payload users with the required orbital state vector data. In addition it is very desirable to minimize the modifications to the spacecraft. Finally, the stationkeeping system should be entirely self-contained (except for initialization) and should not place requirements on either the ground navigation system or the payload during the periods of autonomous operations.

Keeping these ideas in mind, a conceptual "strawman" autonomous stationkeeping system design has been developed. A functional block diagram of this system is presented in Figure 2-7. As may be seen, the functional elements of this system correspond directly to the functional elements of the current ground navigation system.

The strawman system proposes the use of data from separate star sensors, Earth sensors, and sun sensors. The star sensors would be added to the spacecraft. It may also be necessary, in order to achieve the required accuracy, to replace the existing sun sensors (mounted on the solar panels), with a set of body mounted sun sensors. Pending more detailed investigation of its accuracy, it is hoped that the existing Earth sensor can be used. Clearly, sharing the sensors between attitude control and navigation is an attractive option. This will be considered in the design phase.

The sensor data is processed through smoothing, calibration, and editing algorithms to the Orbit Determination program. The primary function of Orbit Determination is, of course, to produce an estimate of the current spacecraft orbit. In addition it may be possible to produce estimates of sensor biases and various propulsion parameters. The additional estimates would be used to both improve the orbit estimation accuracy and support fault detection in other spacecraft systems. For example, it may be possible to estimate the actual magnitudes of the maneuvers applied by the propulsion system and thus provide an independent data source for detecting and isolating propulsion system failures.

Serving both the Orbit Determination function and the Maneuver function is a Trajectory function. The Trajectory function is responsible for providing all predicting estimates of the trajectory. In this way consistency is maintained across the various elements. The trajectory function provides the state estimates required by the payload users and provides sun and moon occultation predictions to attitude control.

The maneuver function takes current orbit estimates from orbit determination and the predicted orbit from trajectory, and determines when maneuvers are required to stay within the specified deadbands. When a maneuver is required, this function first computes the ideal maneuvers and,

using data from the propulsion system, computes the maneuver commands. The commands are transmitted to attitude control for execution.

It must be emphasized that Figure 2-7 represents only a strawman proposal and that analysis has not been conducted on its accuracy characteristics. If the accuracy proves to be inadequate then it may be necessary to employ the more sophisticated sensors such as Madan and Space Sextant.

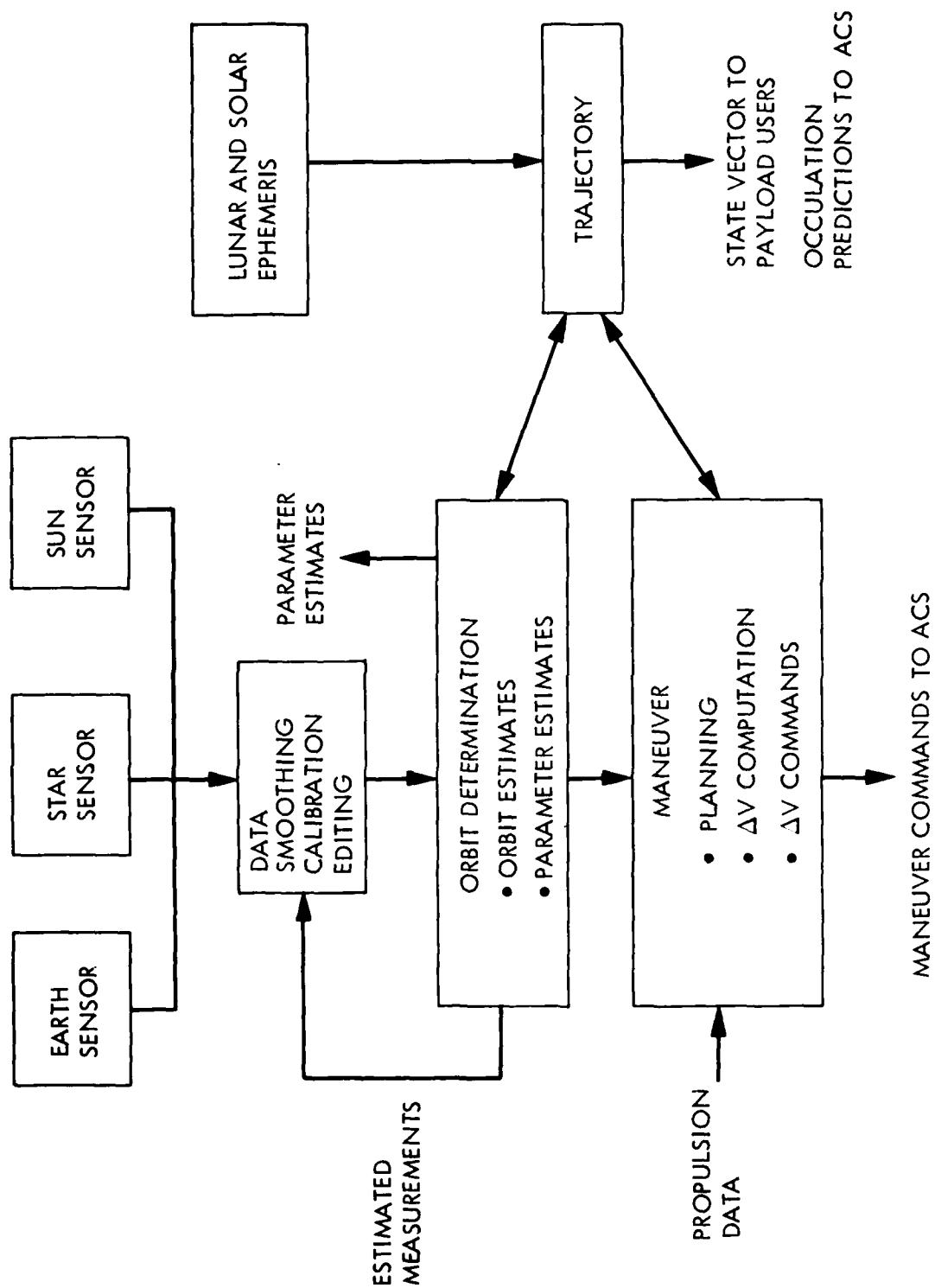


Figure 2-7. Autonomous Stationkeeping Functional Block Diagram

SECTION 3

MANAGE RESOURCES

Figure 3-1 shows a hierarchy of the Resource Management functions.

3.1 AUTONOMOUS OPTIONS FOR POWER RESOURCE MANAGEMENT*

Management of generated power (solar arrays) is largely autonomous. The primary requirements for autonomy of stored energy management and battery life management are to add an on-board capability for battery state assessment.

3.1.1 Manage Generated Power

3.1.1.1 Solar Array (SA) Attitude - Level 5/Category I. Autonomous with respect to sun sensor control.

3.1.1.2 Solar Array Operating Point - Level 0/Category III. Independent control of this function is not attractive for earth orbit applications with stabilized SA temperatures. Major subsystem redesign would be required for a very small payoff.

3.1.2 Manage Stored Energy

3.1.2.1 Sense On-Board Battery Parameters - Level 1/Category I. Sensing functions may be implemented to any extent necessary --functions determined by accuracy and extent of models used in 3.1.2.2.

3.1.2.2 Assess State-of-Charge - Level 0/Category I. Using measured battery parameters and a stored battery state-of-charge model, determine the state of charge. Using stored solar array output and load profile predictions, estimate the state-of-charge during subsequent eclipses. If the state-of-charge trend is downward over several eclipses, the load profile and/or the charge rates must be altered. What may be happening is that the charge rate in use may not completely replenish the battery before its next usage, in which case a higher I-V curve will be requested. Additional choices involve selection of any of the 4 V-T characteristics for determination of taper charge onset, disconnection of the battery charger, and actuation of

*By R. C. Detwiler, T. W. Koerner and G. W. Wester

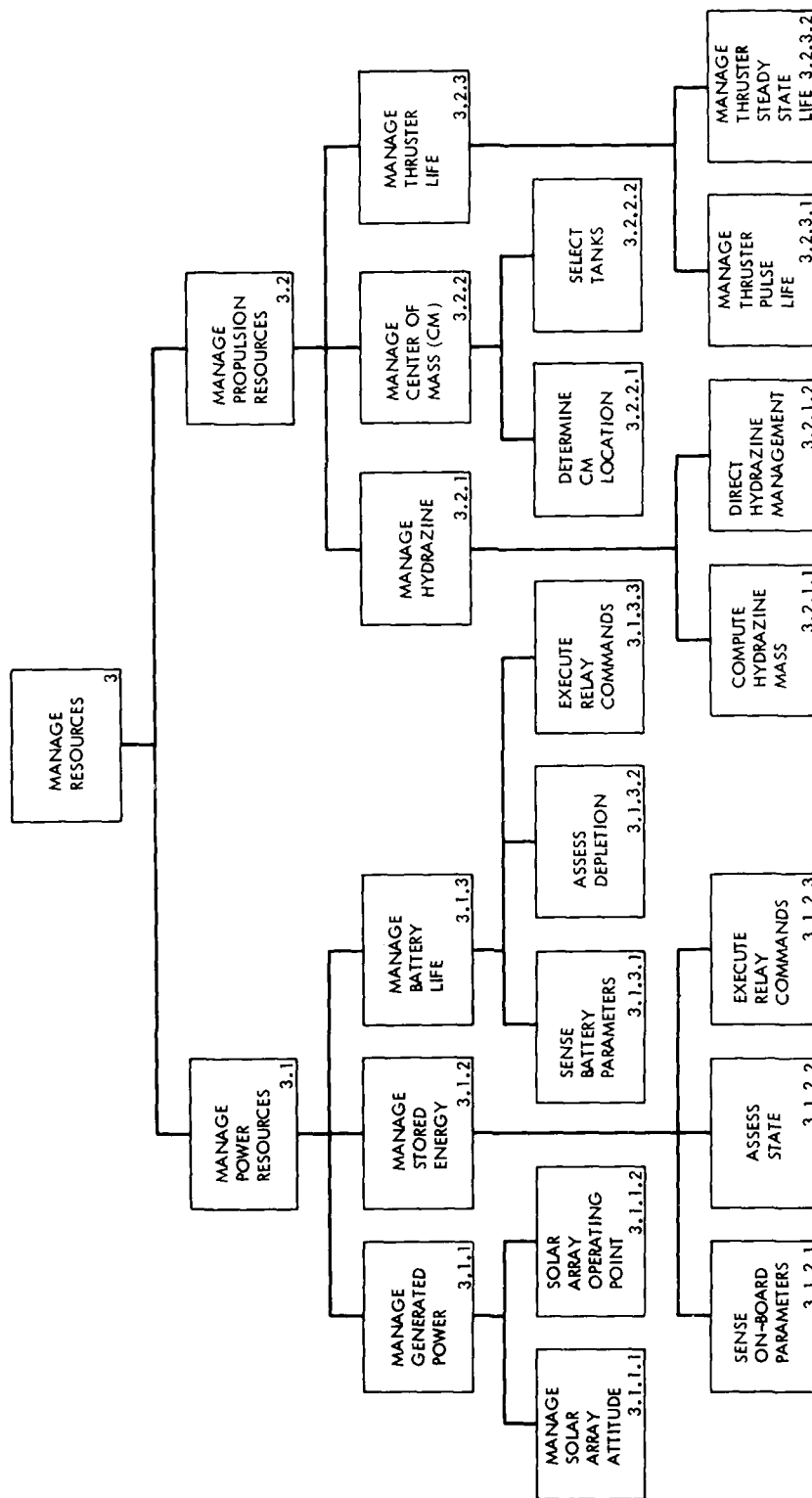


Figure 3-1. Resource Management Functional Hierarchy

either or both battery heaters. Logic diagrams are shown in the DSCS III orbit ops handbook. Since all the information needed for the above decision logic is available through on-board telemetry channels, the logic could probably be implemented on-board with appropriate tolerance detectors and some form of microprocessor. Outputs would have to interface with relay controls that are now activated by ground command. This approach could probably handle most initial battery/charge problems leaving backup action to ground control. To accomplish a reasonably accurate battery state-of-charge estimation on-board would require a fairly sophisticated battery model involving current, voltage, temperature, and possible charge history; a fairly complex computation is implied. Also, a knowledge of the projected load profiles during subsequent eclipses is needed to determine if these loads are too great. Alternatively, a simpler management philosophy can be employed that requires essentially a full recharge after each eclipse; this would simplify the calculation process by requiring load integration over only one eclipse, but might unduly constrain the load profile, particularly if a large load variation from eclipse to eclipse is desired. A great deal depends upon how "tightly" the power subsystem must be designed.

Additional capability required:

- (1) Battery state-of-charge model
- (2) Battery charge history
- (3) Array output predictions
- (4) Load profile predictions
- (5) Load management capability
- (6) Processor capability
- (7) Battery charge relay control

3.1.2.3 Execute Relay Commands - Level 2/Category I. Execute commands for power distribution relays and/or battery charge control relays.

3.1.3 Manage Battery Life

 This function identical to 3.1.2 except that an 80% depth-of-discharge (DOD) should not be exceeded for life maintenance.

3.1.3.1 Sense Battery Parameters - Level 1/Category I.

 See options described under 3.1.2.2.

3.1.3.2 Assess Depletion - Level 0. Voltage readings together with the battery charging model will allow calculation of the DOD during eclipse operations. A comparison with depletion groundrules may decide that system loads are excessive or that power generation/energy storage is inadequate to maintain battery life over a succession of such cycles. This approach requires record keeping of battery voltage over several eclipses and smoothing of the data to see if the trend is downward. Also, if the loads can vary from one eclipse to another, a more sophisticated battery model must be used to estimate the state of charge.

Additional capability required:

- (1) Battery state-of-charge model
- (2) Battery charge history
- (3) Load profile predictions
- (4) Load management capability
- (5) Array profile predictions
- (6) Processing capability for handling the above

3.1.3.3 Execute Relay Commands - Level 2/Category I. Critical functions are redundant as opposed to autonomous.

3.2 AUTONOMOUS OPTIONS FOR PROPULSION RESOURCE MANAGEMENT

All options will require on-board analysis by the propulsion function using sensor data (e.g., telemetry transducers augmented by pulse counts and V firing times from attitude control and navigation. Outputs will be used in decision making by the propulsion, attitude control, navigation and system functions.

3.2.1 Manage Hydrazine resources

3.2.1.1 Compute Hydrazine Mass - Level 2/Category II. The analysis requires pressure and temperature of each tank (available in telemetry). The computations can be verified with estimates based on thruster usage.

3.2.1.2 Direct Hydrazine Management - Level 2/Category II.

3.2.1.2.1 Select Thrusters. Thruster selection to conserve hydrazine by offsetting E/W drift during reaction wheel unloading requires an autonomous navigation capability.

3.2.1.2.2 Select Stationkeeping Strategy. Autonomous reduction of maneuvers to preserve hydrazine would require sophisticated on-board logic to trade-off hydrazine remaining vs. orbit position accuracy. relatively simple priority tables could be stored on the spacecraft in case of a high conflict level.

3.2.2 Manage Center of Mass - Level 2/Category I

Hydrazine tank selection and c.m. control can be automated by additional software and I/O channels for sensor data that is now on the telemetry stream e.g., tank pressure and temperature, and microswitch monitors for feedline latching valves. The software would include analysis functions to determine fuel usage, c.m. migration, and corrections required.

3.2.3 Manage Thruster Life

3.2.3.1 Manage Thruster Pulse Life - Level 2/Category II. Analysis of thruster pulse life requires accumulated count as well as trend (pulse accumulation rate) analysis. Excessive pulse accumulation, which could lead to degradation of performance or the loss of a thruster, would require changes in operating parameters or mission operations to extend thruster life. Addition of pulse counters is required to perform this function. Additional sensors may be desirable for integrity maintenance. Ability to isolate smaller blocks of thrusters (e.g., pairs) would provide an increased depth of fault tolerance at the expense of additional latching valves.

3.2.3.2 Manage Thruster Steady State Life - Level 2/Category III.

Analysis of thruster steady state life requires accumulated firing time and trend (rate of accumulation) analysis. This may not be required on DSCS III as the thruster life requirement can only be exceeded in a worst case scenario for N/S stationkeeping and then only near end-of-life. Autonomous selection of redundant velocity adjust thruster pairs is contingent on having an automated NAV system, and the same additional software in ACS as in 2.2.2.

SECTION 4

AUTONOMY OPTIONS FOR INTEGRITY MAINTENANCE

4.1 AUTONOMOUS SPACECRAFT REDUNDANCY MANAGEMENT OPTIONS* - LEVEL 2/CATEGORY I

Figure 4-1 shows the hierarchy of the integrity maintenance functions. Integrity maintenance is currently a ground-intensive activity. In order to assess options for moving these functions to the spacecraft, design concepts were developed for an integrated approach to on-board integrity maintenance. Options for autonomous integrity maintenance for the individual functions (power, attitude control, thermal control, S/C control and monitoring, propulsion and stationkeeping) are discussed in Sections 4.2-4.7. Their individual functional hierarchies are presented in each section. The most straightforward way of making the DSCS III satellite fault tolerant is to provide an on-board system for managing the spacecraft's redundancy. Two examples of architectural conceptual designs for achieving on-board redundancy management are presented in this section. The candidate design architectures both include a fault-tolerant Redundancy Management Subsystem (RMS) capable of providing fault detection and correction functions for the entire DSCS III spacecraft bus.

The first design architecture extends a concept developed for redundancy management of the TT&C subsystem to redundancy management of other spacecraft functions. Using only an RMS for fault protection processing requires a relatively modest design addition to the DSCS III spacecraft, but its capacity for redundancy management is limited. The RMS uses a unique combination of majority voting, block redundancy and self-generated diagnostic signal techniques to achieve full protection against performance degradation due to internal single-point failures. In addition, valid spacecraft diagnostic data can still be provided to the ground following multiple internal failure occurrences. Being functionally transparent to normal DSCS III on-board operations, addition of the RMS to the spacecraft bus has virtually no impact on existing DSCS III subsystem designs. The estimated mass and power requirements for this RMS are 6 Kg and 8 watts respectively. This "modest" RMS architecture is described in Section 4.1.1.

The second design architecture is capable of increasing the DSCS III on-board autonomy provided by the modest RMS architectural design of Section 4.1.1 through the addition of distributed processing techniques. This capability can be utilized to accommodate a broad range of spacecraft and mission autonomy needs while maintaining a relatively simple multimission RMS design. For this architecture, the RMS I/O module design is modified to include an external supervisory data bus and dedicated reply lines for interfacing with selected subsystems. Each subsystem interface is accomplished through a standard Distributed Processing Unit (DPU). A DPU stores and executes autonomy-related, subsystem-unique software subroutines

*By W. E. Arens and U. S. Lingon

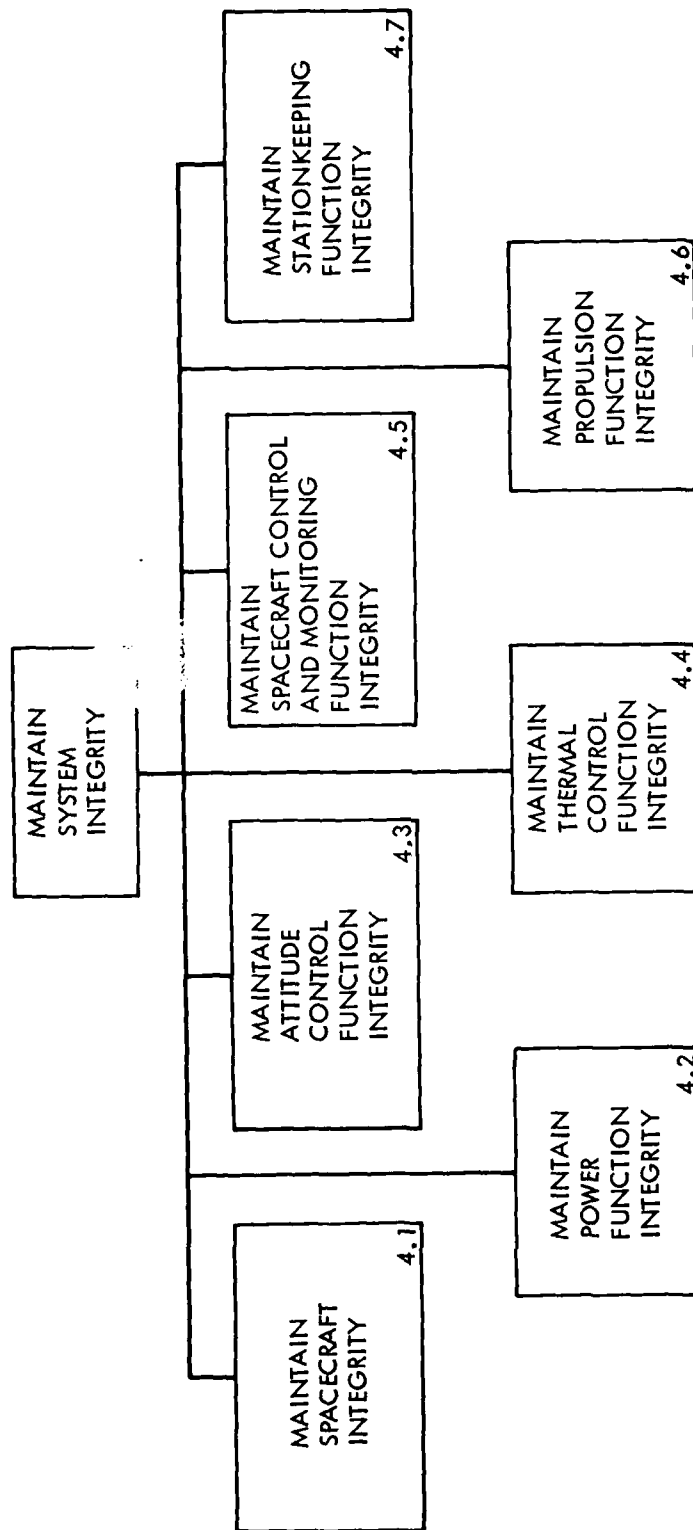


Figure 4-1. Integrity Maintenance Functional Hierarchy

under executive control of the centralized RMS. The RMS interrogates DPU processing results and uses the acquired information to assist in the fault detection and recovery process. The estimated weight and average power for a redundant DPU is 3 kg and 2 watts, respectively. This would be added to the weight and power requirements for the host subsystem. The estimated weight and average power requirements for the modified RMS remain the same as for the modest RMS, i.e., 6 kg and 8 watts respectively.

The second architecture is referred to as the "extensive" RMS design architecture and is described in Section 4.1.2. For ease of reading, some aspects of the "modest" RMS architecture are repeated in the "extensive" RMS architecture discussion.

For the "extensive" RMS system design architecture, a nonvolatile mass storage capability of considerably greater capacity than that provided by the RMS alone appears necessary to support autonomous operation of the entire DSCS III spacecraft for extended periods of time (up to 6 months). Data to be stored would include spacecraft status information, fault history diagnostics and critical software programs. Preliminary estimates indicate requirements to store between 10^8 and 10^9 bits of data. A Data Memory Subsystem (DMS) consisting of two redundant, flight-qualified, radiation-hardened, Galileo Digital Tape Recorders (DTRs) has been defined as a candidate design approach for providing this function. Each DTR would be capable of accommodating a nonvolatile storage capacity of 9×10^8 bits. Using the nonvolatile storage of the RMS as a buffer, DMS usage could be minimized so that the potential for achieving a DTR lifetime of 10 years appears feasible. The average power and weight requirements for the total DMS are estimated to be 3 watts and 18 kg, respectively. This candidate DMS design approach is described in more detail in Section 4.1.3.

Section 4.1.4 presents a specific example of how the Redundancy Management Subsystem could be used. The DSCS III battery high temperature recovery procedure was analyzed (at a conceptual level) to:

- (1) Test the RMS design architecture,
- (2) Provide an example of a fault protection algorithm, and
- (3) Provide a preliminary estimate of computer capabilities required to provide fault protection in one area.

4.1.1 A "Modest" Redundancy Management Design Architecture

A functional description of the DSCS III TT&C Master Telemetry Unit (MTU), Remote Telemetry Unit (RTU), and Command Decoder (CD) is provided in Section 2.4.1 of Volume II. The current method used for DSCS III to achieve fault detection and correction for these functional elements (requiring extensive ground intervention) is described in Section 4.1, Volume II. Section 4.5 of Volume III describes : 1) the use of a Redundancy Management Subsystem

(RMS) for moving current, ground-based, fault tolerance capability to the spacecraft; and 2) the functional characteristics of an RMS capable of making the MTU, RTU, and CD fault tolerant to single-point failures. This section defines the design architecture for an RMS which is 1) inherently fault tolerant to its own internal single-point failures and 2) capable of providing some fault detection and correction functions for the entire spacecraft. The payload functions are not included per se, but the RMS architecture is extendable to a system which could provide payload redundancy management. Section 4.1.1.1 defines the pertinent functional requirements for the RMS. Key assumptions that drive the design architecture and implementation characteristics of the RMS are identified in Section 4.1.1.2. A candidate RMS design architecture, describing pertinent interface and functional characteristics, is defined in Section 4.1.1.3. A possible implementation approach for the defined RMS architecture (including power, weight, and size estimates) is defined in Section 4.1.1.4. This preliminary design definition is intended to be used as a first model for assessing the impact and ultimate feasibility of meeting the fault detection and correction requirements imposed upon the RMS by the existing DSCS III spacecraft subsystems.

4.1.1.1 Functional Requirements. The pertinent functional requirements imposed upon the RMS so that it can provide autonomy for the DSCS III spacecraft bus are defined as follows:

- (1) The RMS shall acquire pertinent health information for each spacecraft subsystem by issuing simulated sensor signals, monitoring the output telemetry stream, and monitoring special-purpose diagnostic responses.
- (2) The RMS shall analyze acquired health information by detecting fault occurrences, isolating fault sources, and defining the required commands to be issued for fault correction.
- (3) The RMS shall generate fault correction commands, each corresponding to an apriori-defined fault condition, by accessing the necessary commands from memory and validating their integrity prior to issuance.
- (4) The RMS shall output validated fault correction commands to the TT&C CD for issuance and execution.
- (5) The RMS shall verify proper execution of fault correction commands by monitoring the state of selected bi-level telemetry measurements.
- (6) The RMS shall store pertinent spacecraft diagnostic information (including time-tagged fault occurrences, fault isolation information, definition of required corrective action, corrective action taken, and results of corrective action) for interrogation by the ground.
- (7) The RMS shall be fault tolerant so that any internal single-point failure will not degrade its performance.

- (8) The RMS shall automatically disable itself from outputting fault correction commands under conditions of multiple failures where its performance is degraded.
- (9) The RMS shall be capable of satisfactorily performing diagnostic activities (fault detection, fault isolation, and definition of required corrective action) and storing pertinent diagnostic results for interrogation by the ground under conditions of multiple failures.

4.1.1.2 Design Assumptions. Significant design assumptions affecting the RMS architectural design and implementation characteristics are defined as follows:

- (1) The RMS will impose no internal design changes to existing DSCS III subsystems.
- (2) The RMS will interface with the TT&C subsystem only.
- (3) The RMS will be transparent to DSCS III on-board operations (simply replacing the ground as the source of fault correction commands).
- (4) The RMS can be overridden and/or disabled by ground command at any time (allowing reversion to a fully intact non-autonomous DSCS III spacecraft).
- (5) The RMS will be implemented using radiation-hardened and flight-qualified Galileo components where applicable (1802 microprocessor, TTC 244 memory chips, and CMOS 4000 series logic chips).
- (6) The RMS will be packaged using Galileo leadless carrier packaging techniques.
- (7) The RMS will be sized to accommodate 8 kilowords of fault-tolerant read/write memory (RWM) and 2 kilowords of fault-tolerant programmable read-only memory (PROM).
- (8) The RMS will be sized to accommodate a fault-tolerant nonvolatile memory capacity of 10^6 bits.

4.1.1.3 Design Architecture. A candidate design architecture for an RMS which is fully capable of meeting the functional requirements of Section 4.1.1.1 in conformance with the design assumptions of Section 4.1.1.2 is described as follows:

4.1.1.3.1 Interface Description. A signal interface block program for the RMS is provided in Figure 4-2. As noted from Figure 4-2, the RMS interfaces only with the DSCS III TT&C subsystem. The spacecraft telemetry stream is

accessed at the output of the TT&C Master Telemetry Unit (MTU) prior to encryption. The RMS can therefore monitor, analyze, and determine the health status of the entire spacecraft (using information extracted from the telemetry stream) with negligible effect on the design or normal operations of the existing DSCS III spacecraft.

Since the RMS represents a functional subsystem of the autonomous DSCS III spacecraft bus, it provides engineering measurements to the TT&C multiplexers so that its own health status can be determined from the telemetry stream. As described in Section 4.5, the RMS also adds diagnostic measurements to the telemetry channel allocation to aid in the fault detection/isolation process for the TT&C subsystem.

Figure 4-2 shows a two-way command interface with the TT&C command decoder (CD). As described in Section 4.5, the RMS sends self-addressed diagnostic commands to the CD so that it can evaluate the functional integrity of the CD. It also sends fault correction commands to the CD for issuance to the appropriate spacecraft subsystems following detection and isolation of faults. Commands routed to the CD from the RMS are identical to plain-text commands received by the CD from the ground. Therefore, the RMS simply replaces the ground as a command source resulting in no impact to existing DSCS III spacecraft subsystem designs.

Reference to Figure 4-2 also shows the existence of an interface between the TT&C CD and RMS for routing commands from the ground to the RMS. This allows the ground to override, reconfigure, and even disable the RMS if desired. Since the RMS is virtually transparent to the remainder of the spacecraft, it can be totally disabled by ground command with no impact on the existing nonautonomous DSCS III design or operational capabilities.

An additional interface defined in Figure 4-2 allows pertinent telemetry and diagnostic data, stored by the RMS during periods of autonomous operation of the spacecraft, to be read from storage and transferred to the TT&C RF equipment (RFE) for encryption, carrier modulation, and transmission to the ground. Interrogation of this stored data is initiated by a ground command through the appropriate TT&C CD and RMS command interface defined in Figure 4-2.

4.1.1.3.2 Design Description.

4.1.1.3.2.1 Real Time Fault Protection. A block diagram for a candidate fault-tolerant RMS design architecture is provided in Figure 4-3. Figure 4-3 shows three identical Redundancy Management Modules (RMMs) each interfacing with two identical Input/Output (I/O) modules. The block diagram for an RMM is given in Figure 4-4. As noted from Figure 4-4, an RMM interfaces with the I/O modules of Figure 4-3 via internal supervisory and reply busses. Traffic on both the supervisory and reply busses of an RMM is controlled by a central processor unit (CPU). The CPU, in effecting bus control, provides both timing and digital processing functions.

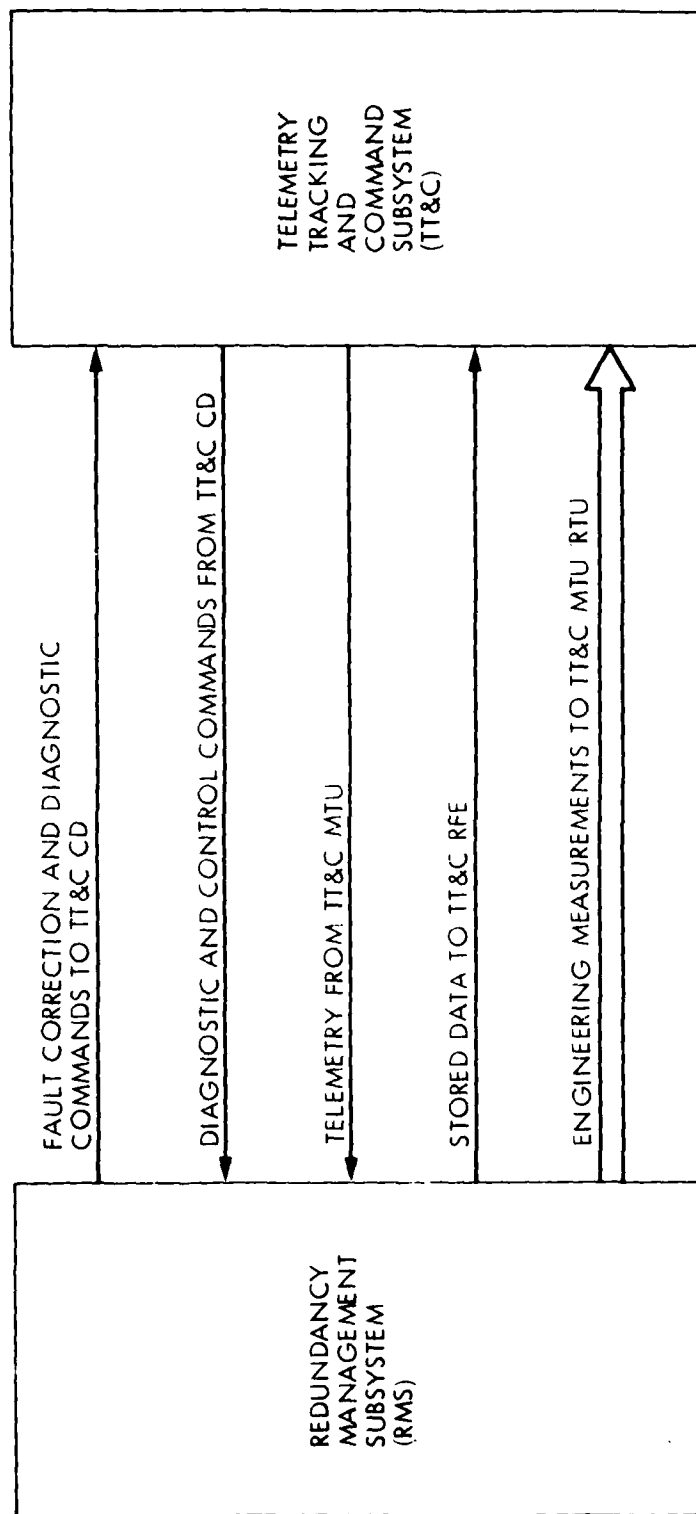


Figure 4-2. RMS Interface Block Diagram

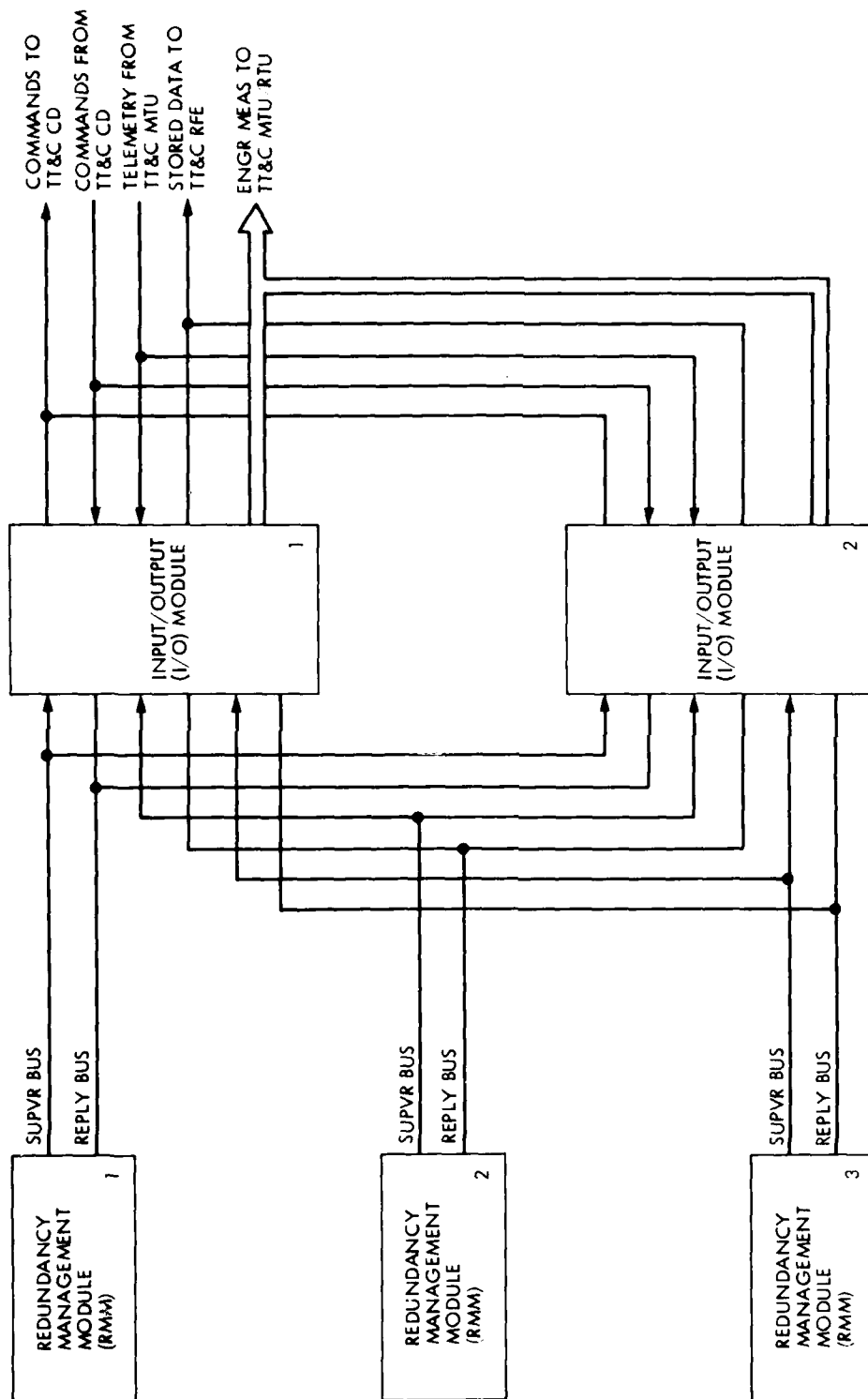


Figure 4-3. RMS Block Diagram

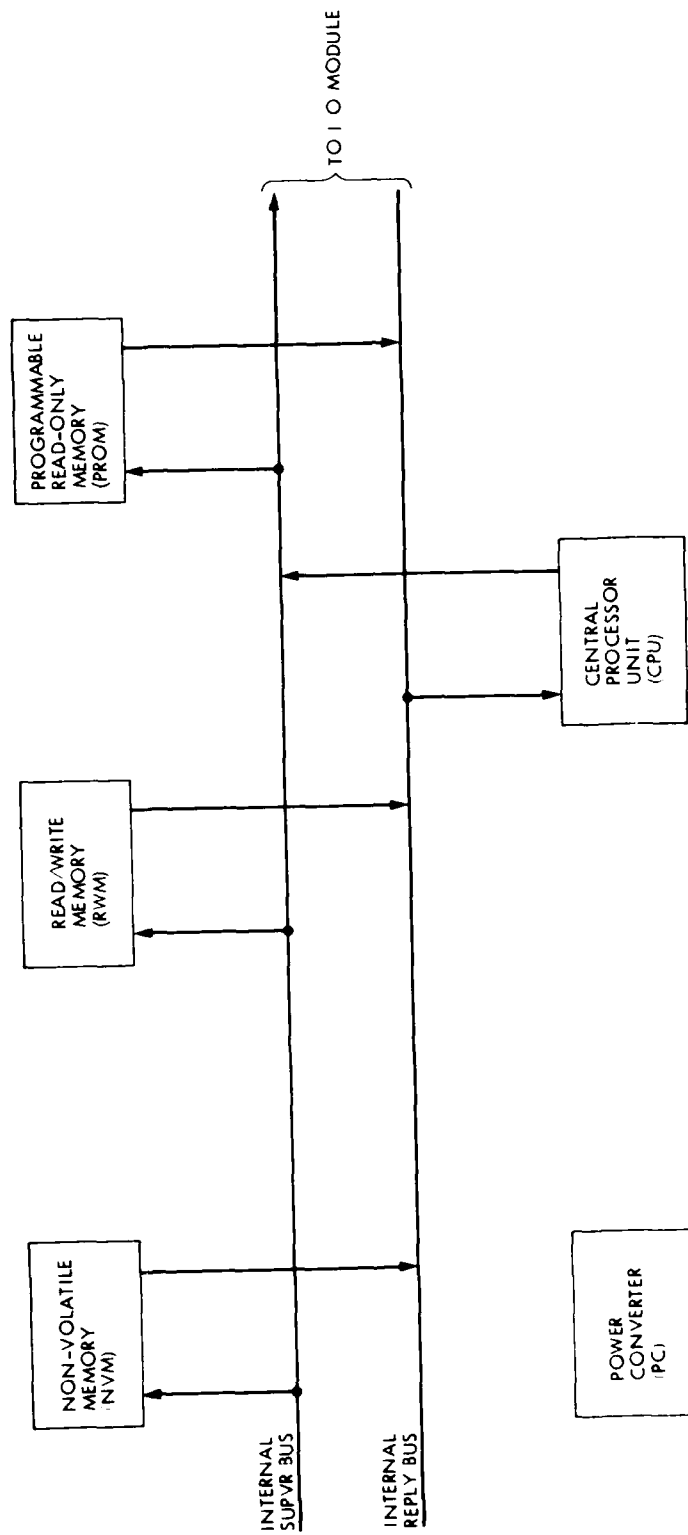


Figure 4-4. RMM Block Diagram

Referring to Figure 4-4, a programmable read-only memory (PROM), a volatile read-write memory (RWM), and a nonvolatile memory (NVM) are all connected to the CPU through the common bus structure. The PROM stores executive software and RWM command address tables. The RWM stores fault detection software routines and fault correction commands. The RWM also buffers pertinent telemetry and diagnostic data prior to long-term storage. The NVM provides long term storage of pertinent telemetry and diagnostic data for subsequent transmission to earth. The NVM also stores critical software routines for reloading the RWM in the event of RWM memory loss resulting from, for example, power interruptions.

As indicated in Figure 4-4, all instructions for data transfer to and from other blocks (including the I/O module) are issued over the supervisory bus from the CPU. Such supervisory commands contain the proper block address so that only the addressed block will respond over the reply bus at any given time. Any external commands coming from the I/O module would be interrogated and requested by the CPU via the supervisory bus and transferred to the CPU for execution via the reply bus.

Referring to Figure 4-3, both I/O modules are individually powered and active. Any communications from an RMM CPU over a supervisory bus will normally be addressed to only one of the two I/O modules. Therefore, only the addressed I/O module will be able to transfer information to the RMS or TT&C subsystem. If a failure occurs in the selected I/O module, an RMM CPU detects this via several indications (lack of response from periodically issued diagnostic self-addressed commands to the TT&C CD, improper telemetry response to diagnostic data, etc.) and merely addresses further communications over the supervisory bus to the other I/O module.

A block diagram for an I/O module is given in Figure 4-5. As noted from Figure 4-5, commands addressed to a given I/O module will come from each RMM CPU over their respective supervisory busses to a majority voter unit. Therefore, fault correction and diagnostic commands to be issued externally through the command output unit will require that agreement of at least two of the three CPU outputs be achieved. If one RMM fails, RMS operation will be unimpaired since agreement will be attained from the remaining two RMM outputs. If failures occur in two or more RMMs, the majority vote agreement would not be achieved and the issuance of fault correction and diagnostic commands from the RMS would be inhibited unless RMS reconfiguration is commanded from the ground.

Referring to Figure 4-5, incoming commands from the TT&C CD can be routed to all RMMs simultaneously or any selected RMM via the designated reply busses. An application where a command would be addressed to a single RMM is a request from the ground to interrogate diagnostic data stored in the NVM of a given RMM. The CPU of the addressed RMM would execute readout of its NVM. The data would be routed over the RMM reply bus, on a noninterference basis, to the stored data output unit of the designated I/O module. Readout of the diagnostic data from the NVM of each remaining RMM could also be requested individually via properly addressed ground commands. Since the telemetry stream from the TT&C CD is available to all RMM CPUs via individual reply lines, this approach of individual readout of RMM diagnostic data could allow for multiple failures, i.e., two of the three RMMs could fail and valid fault detection, isolation, and required correction information could still be

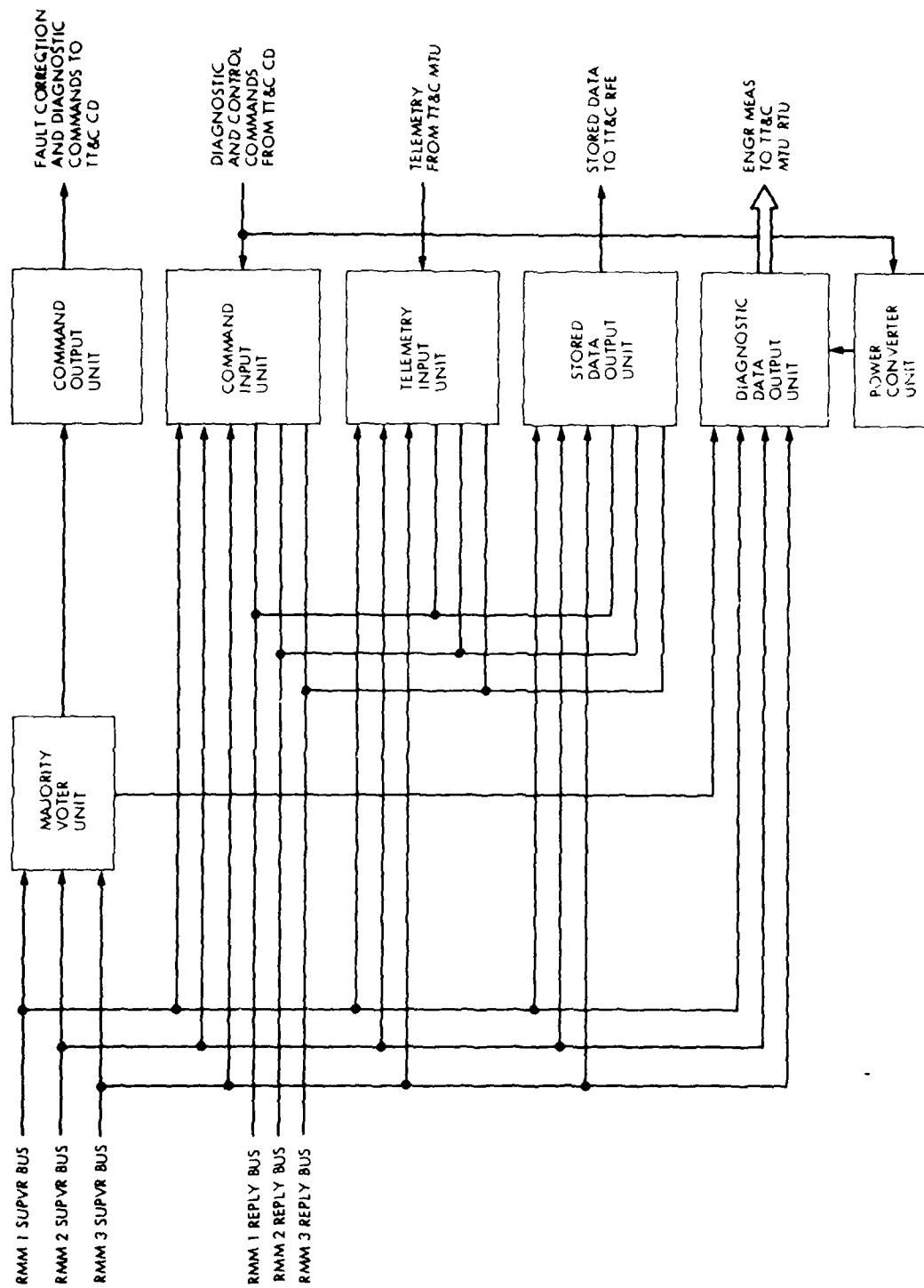


Figure 4-5. I/O Module Block Diagram

transferred to the ground by the good RMM. This not only would benefit mission operations activities on the ground related to fault detection and correction, it would also provide information as to which RMMs had failed. Furthermore, the design of the I/O module could be effected such that the majority voter unit could be bypassed with any selected supervisory bus via a single ground command. This would allow an undegraded fault correction capability to be reinstated with only one of the three RMMs operational. Since the selected RMM CPU could detect a failure in a given I/O module and subsequently address the remaining I/O module, multiple failures, including total failure of everything but one RMM and one I/O module, could theoretically allow undegraded spacecraft autonomy functions to still be achieved.

As noted in Figure 4-5, a discrete command could be sent directly from the TT&C CD to a power converter in the I/O module via a dedicated line to switch power on and off. This would allow complete disablement of one or both I/O modules via ground command. The net result would allow reversion of the DSCS III spacecraft to the nonautonomous condition with no degradatory effects with respect to current design and operational characteristics. This aspect of the defined architecture should greatly reduce the initial risk of integrating autonomy into the existing DSCS III flight-qualified system design.

4.1.1.3.2.2 Fault Protection Traceability. This section describes an RMS methodology for detecting faults and generating diagnostic data for fault history traceability using the output telemetry data stream from the TT&C subsystem MTU.

4.1.1.3.2.2.1 Telemetry Acquisition. As described in Section 4.5, the RMS accesses the unencrypted telemetry output data stream via dedicated lines from the TT&C subsystem MTU. Referring to Figure 4-3, it is noted that the telemetry data is routed from an RMS I/O module to each of three Redundancy Management Modules (RMMs). An RMM, defined in Figure 4-4, monitors the incoming telemetry data stream for measurement anomaly occurrence. The RMM CPU accesses a TT&C subsystem master telemetry frame from the telemetry input unit of the I/O module of Figure 4-5 via a request over its supervisory bus. The measurement data for the master telemetry frame is routed to the CPU via its dedicated reply bus.

The CPU selects apriori-defined measurements from the TT&C master telemetry frame to form a fault detection data frame to be used for fault detection purposes. The 7680 words in a TT&C subsystem master telemetry frame contain 320 analog measurements (320 8-bit words when digitized), 490 bi-level measurements (70 8-bit words), 2 23-bit serial digital measurements (8 8-bit words), and 16 8-bit serial-digital measurements. Therefore, all of the different measurements accommodated by a TT&C subsystem master telemetry frame can be represented by 414 8-bit words. Since all of these will not be used for fault detection and correction purposes, a realistic fault detection data frame size is assumed to be 256 8-bit words.

4.1.1.3.2.2.2 Fault Detection and Correction. Prior to acquiring each new master telemetry frame, measurement data corresponding to that for the previous fault detection data frame is present in the read/write memory (RWM) of Figure 4-4. Current fault detection data frame measurements, being accessed from the incoming master telemetry frame, are compared with corresponding measurements from the preceding frame on a bit-by-bit basis using half-adder logic in the CPU. If there is no unacceptable change in a compared measurement, the data from the previous fault detection data frame, for that measurement, is retained in the RWM. If there is an unacceptable change, representing an anomaly, the CPU replaces the previous measurement with the current measurement in the fault detection data frame stored in the RWM. Therefore, the fault detection data frame in the RWM is continuously updated. Also, the current value of the measurement that has undergone an unacceptable change is time tagged and routed to a selected address in the RWM for interim storage prior to transfer to the nonvolatile memory (NVM) for long-term storage. Sensitivity of what constitutes an unacceptable change is based upon which significant digit of the 8-bit telemetry measurement word is selected apriori as the least significant bit for responding to bit comparison disagreements.

Following anomaly identification the CPU executes appropriate software routines, as described in Section 4.1.1.3.2.1, above, to effect fault isolation and correction. Pertinent diagnostic data words, with appropriate header identification, are generated by the CPU to identify the source of the fault, the corrective action taken, and the results of the corrective action. These data words are also routed to the RWM for interim storage.

In addition to 1) time-tagged measurements which reflect changes indicative of fault occurrences and 2) fault isolation and correction data, a complete fault detection data frame is also transferred to the RWM for interim storage at periodic intervals. The periodic storage of an updated fault detection data frame validates the integrity of and provides an updated reference for the specific fault diagnostic data.

4.1.1.3.2.2.3 Long-Term Storage. The fault detection and correction data described above is collected in the RWM until a representative data block size has been accumulated. The data block is then transferred, in response to a supervisory bus request from the CPU, to the NVM for long-term storage. It will remain in storage until interrogated by means of a ground read-out command.

It should be emphasized that the quantity of stored fault detection and correction data is very small compared with the total incoming telemetry data. For instance, periodic storage of a complete fault detection data frame may be typically accomplished only once every six hours. Assuming 2048 bits per frame (256 8-bit words), this would require approximately 0.5×10^6 bits of storage for a period of 60 days. Assuming 1) the remainder of the data stored in the NVM for interrogation by the ground would represent only specific fault diagnostic data, and 2) the duty cycle of fault occurrences would be extremely low, then a NVM capacity of 10^6 bits should be sufficient.

4.1.1.3.2.2.4 Other Considerations. Since the updated measurements from the previous telemetry frame are always available when a measurement anomaly is detected, a diagnostic history of a fault occurrence is possible. The frequency response of such a history could be significantly increased if the measurement sampling rate of the TT&C subsystem were proportionally increased during periods of autonomous operation. Since only the number and type of fault occurrences affect the quantity of diagnostic data that must be stored by the RMS, such a sampling rate increase would create little impact on the RMS design requirements. A telemetry rate increase option should be possible for future DSCS III TT&C subsystem designs by providing a higher clock rate for the autonomous mode.

Another consideration for improved RMS diagnostic performance would be to store additional telemetry formats in the TT&C subsystem MTU PROM to provide measurement sampling selectability tailored to apriori-defined diagnostic needs. For instance, having all measurements required for an RMS fault detection frame contained in each TT&C subsystem MTU main frame (30 main frames per master frame) would be highly desirable.

4.1.1.4 Implementation Considerations. A candidate implementation of the RMS design architecture defined in Section 4.1.1.3 commensurate with the design assumptions of Section 4.1.1.2 is described as follows:

4.1.1.4.1. Redundancy Management Module (RMM). The RMM of Figure 4-4 consists of five discrete functional blocks (CPU, PROM, RWM, NVM, and PC). The central processing unit (CPU) would be implemented using an 1802 microprocessor. The CPU peripheral direct memory access (DMA), timing, and control logic circuits would be implemented using CMOS 4000 series logic chips. Both the 1802 microprocessor and CMOS 4000 series logic chips have been radiation hardened and flight qualified on the JPL Galileo program.

The programmable read-only memory (PROM) would be implemented using a commercially available radiation-hardened, flight-qualified unit similar to that already included in the DSCS III TT&C MTU. The RMS PROM would be sized for 2 kilowords (8 bits/word) of memory which should be accommodated by a single chip.

The read-write memory (RWM) would be implemented using 256- x 4-bit TTC 244 CMOS memory chips which have also been radiation hardened and flight qualified on the JPL Galileo program. It would be sized for 8 kilowords (8 bits/word) of memory. This would require about 136 chips. Ninety-six of the 136 memory chips would be required for storage while the remaining 40 chips would be used for driver and decoding functions.

The nonvolatile memory (NVM) would be implemented using a commercially available radiation-hardened 10^6 bit bubble memory chip. The NMOS chips currently used in commercially developed bubble memory units (for memory control, timing, and driver circuits) would be replaced with radiation-hardened, flight-qualified CMOS and bi-polar chips.

The power converter (PC) would be implemented using radiation-hardened, flight-qualified components and circuit designs from the JPL Galileo program.

Packaging of the individual CPU, PROM, RWM, NVM, and PC functional blocks would be accomplished using Galileo leadless carrier packaging techniques. The resultant functional blocks would be housed and interconnected in a standard Galileo subchassis (41.3 cm x 18.4 cm) should be adequate to accommodate an entire RMM.

4.1.1.4.2 Input/Output (I/O) Module. The I/O module of Figure 4-5 consists of seven, discrete, functional blocks. All of these blocks would be implemented using radiation-hardened, flight-qualified components from the JPL Galileo program. The logic circuits and output buffers would use CMOS 4000 series logic chips. The power converter design and implementation would be identical to that used for an RMM.

As for an RMM, packaging of the individual functional blocks of Figure 4-5 would be accomplished using Galileo leadless carrier packaging techniques. The functional blocks for both I/O modules would be housed and interconnected on one side of a standard Galileo subchassis.

4.1.1.4.3 Power, Weight, and Size. The estimated power, weight, and size requirements for the RMS defined herein are provided in Table 4-1. Referring to Table 4-1, each RMM of Figure 4-3 is estimated to require an average power of 2 watts for a total average power of 6 watts for three RMMs. The total peak power for all three RMMs should not exceed 12 watts. This could only occur when and if data were transferred to and from nonvolatile memories in all RMMs simultaneously. Each I/O module is estimated to require an average and peak power of 1 watt for a total power of 2 watts for the two modules of Figure 4-3. As noted from Table 4-1, the total estimated average power for the RMS is therefore 8 watts.

The weight of the RMS is based upon the use of two Galileo subchassis each weighing 1 kg. Since an RMM is mounted on one side of a Galileo subchassis, two subchassis are required to house the three RMMs of Figure 4-3. The three RMMs therefore add 3 kg to the 2 kg required by the two subchassis housings. The two I/O modules of Figure 4-3, each weighing 0.5 kg (including interconnection boards), are mounted on the remaining empty side of one of the subchassis. This results in a total estimated RMS weight of 6 kg.

The two Galileo subchassis used to house the RMS each measure 41.3 cm x 18.4 cm x 4.5 cm for a per unit volume of 3420 cm³. The total estimated volume required by the RMS is therefore 6840 cm³.

Table 4-1. RMS Implementation Characteristics

ELEMENT	UNIT POWER watts	UNIT WEIGHT kg	UNIT VOLUME cm ³	QUANTITY	TOTAL POWER watts	TOTAL WEIGHT kg	TOTAL VOLUME cm ³
RMM	2	1	—	3	6	3	—
I/O	1	0.5	—	2	2	1	—
SUBCHASSIS	—	1	3420	2	—	2	6840
TOTALS	—	—	—	—	8	6	6840

4.1.2 An Extensive Redundancy Management Design Architecture

The previous section (4.1.1) described a candidate design architecture for a "modest" Redundancy Management Subsystem (RMS) design which is 1) inherently fault tolerant to its own internal single point failures and 2) capable of providing some on-board fault detection and correction functions for the DSCS III spacecraft bus. Section 4.1.2 describes a candidate system design architecture for increasing the fault detection and correction capability of the defined RMS through incorporation of distributed processing techniques.

Section 4.1.2.1 defines the pertinent functional requirements imposed upon the architecture. Key assumptions that drive the design and implementation characteristics of the architecture are identified in Section 4.1.2.2. The pertinent interface and functional characteristics of the architecture are defined in Section 4.1.2.3. Sections 4.1.2.4 and 4.1.2.5 describe candidate designs for the major hardware elements of the architecture. Finally, implementation characteristics of the hardware elements of the architecture (including power, weight, and size estimates) are discussed in Section 4.1.2.6.

This preliminary definition of a candidate system design architecture for increasing the on-board autonomy of the existing DSCS III spacecraft bus represents only one of several possible ways of approaching the problem. It should be evaluated in comparison with other candidate architectures to determine its relative suitability for meeting the DSCS III mission needs of the customer.

4.1.2.1 Functional Requirements. A system design architecture is proposed for consideration which would use a centralized Redundancy Management Subsystem (RMS), similar to that described in Section 4.1.1, to provide executive control of subsystem-assigned Distributed Processing Units (DPUs) through a common bus structure. The pertinent functional requirements imposed upon the subject system design architecture so that it can provide autonomy for the DSCS III spacecraft bus are defined as follows:

- (1) A DPU shall acquire health information from its host subsystem by monitoring selected subsystem sensor signals via dedicated lines.
- (2) A DPU shall store software subroutines required to analyze functional performance and determine needs unique to its host subsystem.
- (3) A DPU shall execute selected internally-stored software subroutines only upon receipt of appropriate executive-level supervisory bus commands from the RMS.
- (4) A DPU shall provide processed, subsystem-unique health information to the RMS upon request by the RMS.

- (5) The RMS shall acquire health information for spacecraft subsystems from 1) subsystem DPU reply line responses to RMS supervisory bus commands, 2) the output telemetry stream from the TT&C MTU, and 3) special purpose diagnostic responses to self-addressed commands via dedicated lines from the TT&C command decoder.
- (6) The RMS shall analyze acquired subsystem health information by detecting fault occurrences, isolating fault sources, and defining the required commands to be issued for fault correction.
- (7) The RMS shall generate subsystem fault correction commands, each corresponding to an apriori-defined fault condition, by accessing the necessary commands from memory and validating their integrity prior to issuance.
- (8) The RMS shall output validated fault correction commands to the TT&C subsystem CD for issuance to and execution by the appropriately addressed subsystems.
- (9) The RMS shall verify proper execution of fault correction commands by monitoring the state of selected bi-level measurements received from 1) subsystem DPUs via dedicated reply lines and 2) the output telemetry stream from the TT&C subsystem MTU.
- (10) The RMS shall store pertinent spacecraft diagnostic information (including time-tagged fault occurrences, fault isolation information, definition of required corrective action, corrective action taken, and results of corrective action) for interrogation by the ground.
- (11) The RMS shall be capable of loading the memories of all DPUs connected to its supervisory bus.
- (12) The RMS shall be inherently fault tolerant so that any internal single-point failure will not degrade its performance.
- (13) The RMS shall automatically disable itself from outputting fault correction commands under conditions of multiple failures where its performance is degraded.
- (14) The RMS shall be capable of satisfactorily performing diagnostic activities (fault detection, fault isolation, and definition of required corrective action) and storing pertinent diagnostic results for interrogation by the ground under conditions of multiple failures.
- (15) The RMS shall be capable of performing more limited, but useful, fault correction and detection functions for a

subsystem even in the presence of a complete loss of the RMS bus interface with a DPU for that subsystem.

- (16) The system design shall be capable of efficiently accommodating a broad range of autonomy processing needs with minimal effect on the RMS design through selective allocation of DPUs.

4.1.2.2 Design Assumptions. Significant assumptions affecting the design and implementation characteristics of the subject system design architecture are defined as follows:

- (1) The RMS and subsystem DPUs will impose no significant internal design changes to existing DSCS III subsystems.
- (2) A DPU will receive health information from, but provide no fault correction commanding to, a host subsystem which is currently part of the existing DSCS III design.
- (3) The RMS will route all fault correction commands required by spacecraft bus subsystems through the TT&C subsystem for issuance.
- (4) The RMS and subsystem DPUs will be transparent to normal DSCS III on-board operations.
- (5) The RMS can be overridden and/or disabled by ground command at any time (allowing reversion to a fully intact nonautonomous DSCS III spacecraft).
- (6) The RMS and subsystem DPUs will be implemented using radiation-hardened, flight-qualified Galileo components where applicable (1802 microprocessor, TTC 244 memory chips, and 4000 series logic chips).
- (7) The RMS and subsystem DPUs will be packaged using Galileo leadless carrier packaging techniques.
- (8) The RMS and each subsystem DPU will be sized to accommodate 8 kilowords of fault-tolerant read/write memory (RWM) and 2 kilowords of fault-tolerant programmable read-only memory (PROM).
- (9) A subsystem DPU will be capable of accommodating 64 analog and/or hi-level measurement channels from its host subsystem.
- (10) The RMS will be sized to accommodate a fault-tolerant non-volatile memory capacity of 10^6 bits.

4.1.2.3 System Design Architecture. A candidate system design architecture which is capable of meeting the functional requirements of Section 4.1.2.1 in conformance with the design assumptions of Section 4.1.2.2 is described in Figure 4-6. Referring to Figure 4-6, spacecraft fault detection and correction functions are controlled by an on-board Redundancy Management Subsystem (RMS) which interfaces directly with the DSCS III TT&C subsystem. The spacecraft telemetry stream is accessed at the output of the TT&C Master Telemetry Unit (MTU) prior to encryption. The RMS can therefore monitor, analyze, and determine the health status of the entire spacecraft (using information extracted from the telemetry stream) with negligible effect on the design or normal operations of the existing DSCS III spacecraft.

Since the RMS represents a functional subsystem of the autonomous DSCS III spacecraft bus, it provides engineering measurements to the TT&C multiplexers so that its own health status can be determined from the telemetry stream. As described in Section 4.1.1, the RMS also adds diagnostic measurements to the telemetry channel allocation to aid in the fault detection/isolation process for the TT&C subsystem.

Figure 4-6 shows a two-way command interface with the TT&C command decoder (CD). As described in Section 4.1.1, the RMS sends self-addressed diagnostic commands to the CD so that it can evaluate the functional integrity of the CD. It also sends fault correction commands to the CD for issuance to the appropriate spacecraft subsystems following detection and isolation of faults. Commands routed to the CD from the RMS are identical to plain-text commands received by the CD from the ground. Therefore, the RMS simply replaces the ground as a command source, resulting in no impact to existing DSCS III spacecraft subsystem designs.

Reference to Figure 4-6 also shows the existence of an interface between the TT&C CD and RMS for routing commands from the ground to the RMS. This allows the ground to override, reconfigure, and even disable the RMS if desired. Since the RMS is virtually transparent to the remainder of the spacecraft, it can be totally disabled by ground command with no impact on the existing nonautonomous DSCS III design or operational capabilities.

An additional TT&C subsystem interface defined in Figure 4-6 allows pertinent telemetry and diagnostic data, stored by the RMS during periods of autonomous operation of the spacecraft, to be read from storage and transferred to the TT&C RF equipment (RFE) for encryption, carrier modulation, and transmission to the ground. Interrogation of this stored data is initiated by a ground command through the appropriate TT&C CD and RMS command interface defined in Figure 4-6.

As noted from Figure 4-6, the RMS can also interface with other spacecraft subsystems through a common bus. This interface is accomplished through use of a standard interface module designated as a Distributed Processing Unit (DPU). Use of a DPU and the RMS bus is optional based upon a particular subsystem's needs. If used, the DPU becomes part of the host subsystem's responsibility.

A DPU interfaces with the bus through a standard bus adapter circuit. The DPU interface with a host subsystem, which is currently part of the existing DSCS III design, is limited to receipt of specified measurement signals deemed useful for determining the requirements of and effecting autonomy functions unique to that subsystem. The DPU contains signal

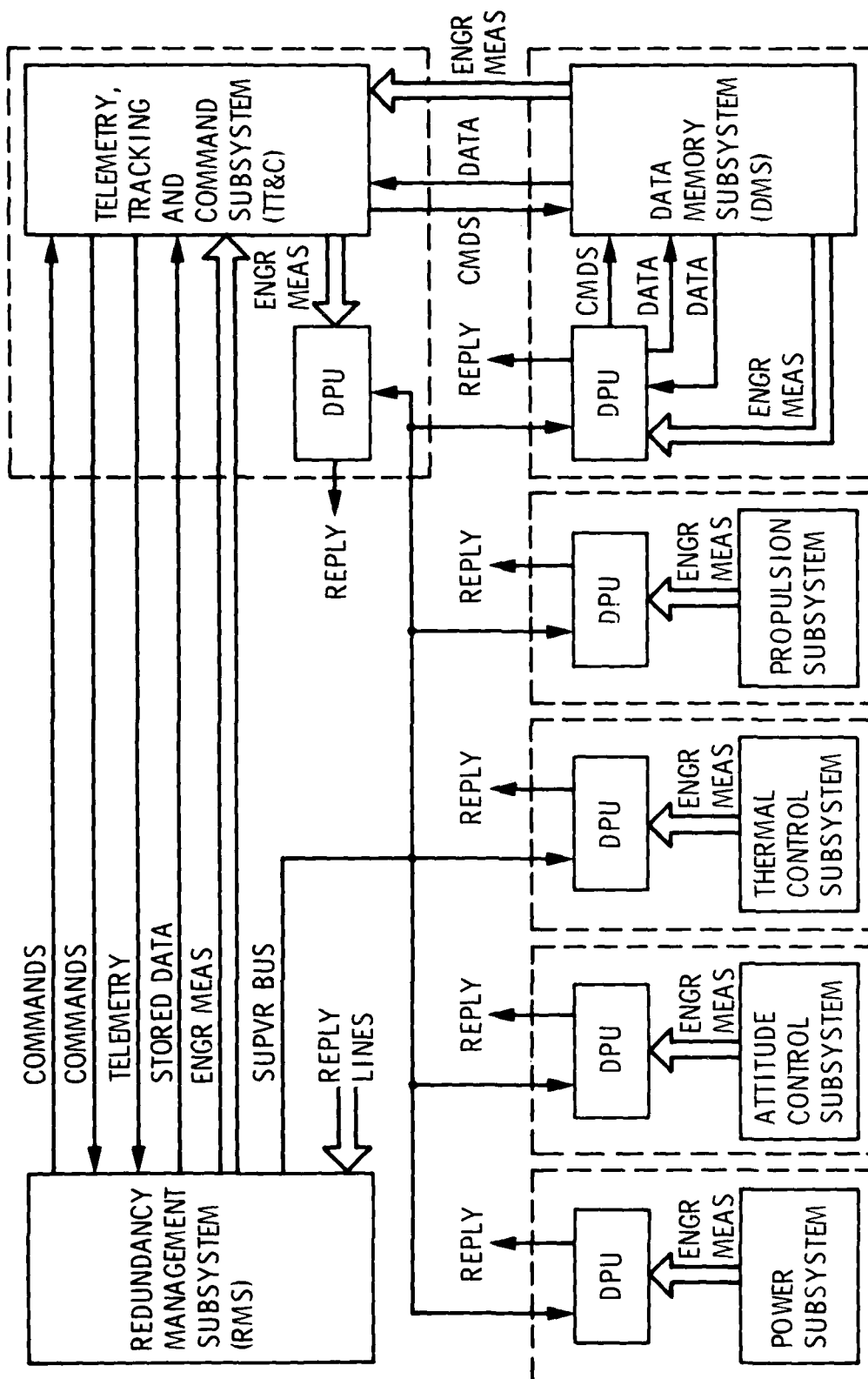


Figure 4-6. System Design Architecture

conditioning, memory, and compute capability so that it can use the incoming information to determine subsystem-unique health status and needs independent of the RMS telemetry stream monitoring and other subsystem activities. DPUs are sequentially interrogated by the RMS via the supervisory bus on a near-continuous basis. DPU status results are returned to the RMS via dedicated reply lines.

It should be noted from Figure 4-6 that, with the exception of the DMS (which represents a new subsystem), a DPU does not effect direct control over its host subsystem. DPU results are provided to the RMS as another source of information. Issuance of fault correction commands is accomplished through the TT&C subsystem interface. This architectural feature maintains the characteristics of transparency to normal DSCS III operations. Although each DPU is considered part of its host subsystem, it really functions as a tool of the RMS.

The use of DPUs allows health monitoring, computations, etc. to be simultaneously accomplished for each subsystem in parallel under executive control of the RMS. Therefore, when compared with the nondistributed architecture of Section 4.1.1, increased and more efficient processing capability for performing fault detection and diagnostic functions can be realized. Furthermore, through reallocation of DPUs, a relatively simple fault-tolerant RMS design can be maintained for system autonomy needs covering a wide range of capability.

4.1.2.4 RMS Design Description. A block diagram for a candidate fault-tolerant RMS design is provided in Figure 4-7. Figure 4-7 shows three identical Redundancy Management Modules (RMMs) each interfacing with two identical Input/Output (I/O) modules. The block diagram for an RMM is given in Figure 4-8. As noted from Figure 4-8, an RMM interfaces with the I/O modules of Figure 4-7 via internal supervisory and reply busses. Traffic on both the supervisory and reply busses of an RMM is controlled by a central processor unit (CPU). The CPU, in effecting bus control, provides both timing and digital processing functions.

Referring to Figure 4-8, a programmable read-only memory (PROM), a volatile read-write memory (RWM), and a nonvolatile memory (NVM) are all connected to the CPU through the common bus structure. The PROM stores executive software and RWM command address tables. The RWM stores fault detection software routines and fault correction commands. The RWM also buffers pertinent telemetry and diagnostic data prior to long-term storage. The NVM provides long-term storage of pertinent telemetry and diagnostic data for subsequent transmission to earth. The NVM also stores critical software routines for reloading the RWM in the event of RWM memory loss resulting from, for example, power interruptions.

As indicated in Figure 4-8, all instructions for data transfer to and from other blocks (including the I/O module) are issued over the supervisory bus from the CPU. Such supervisory commands contain the proper block address so that only the addressed block will respond over the reply bus at any given time. Any external commands coming from the I/O module would be interrogated and requested by the CPU via the supervisory bus and transferred to the CPU for execution via the reply bus.

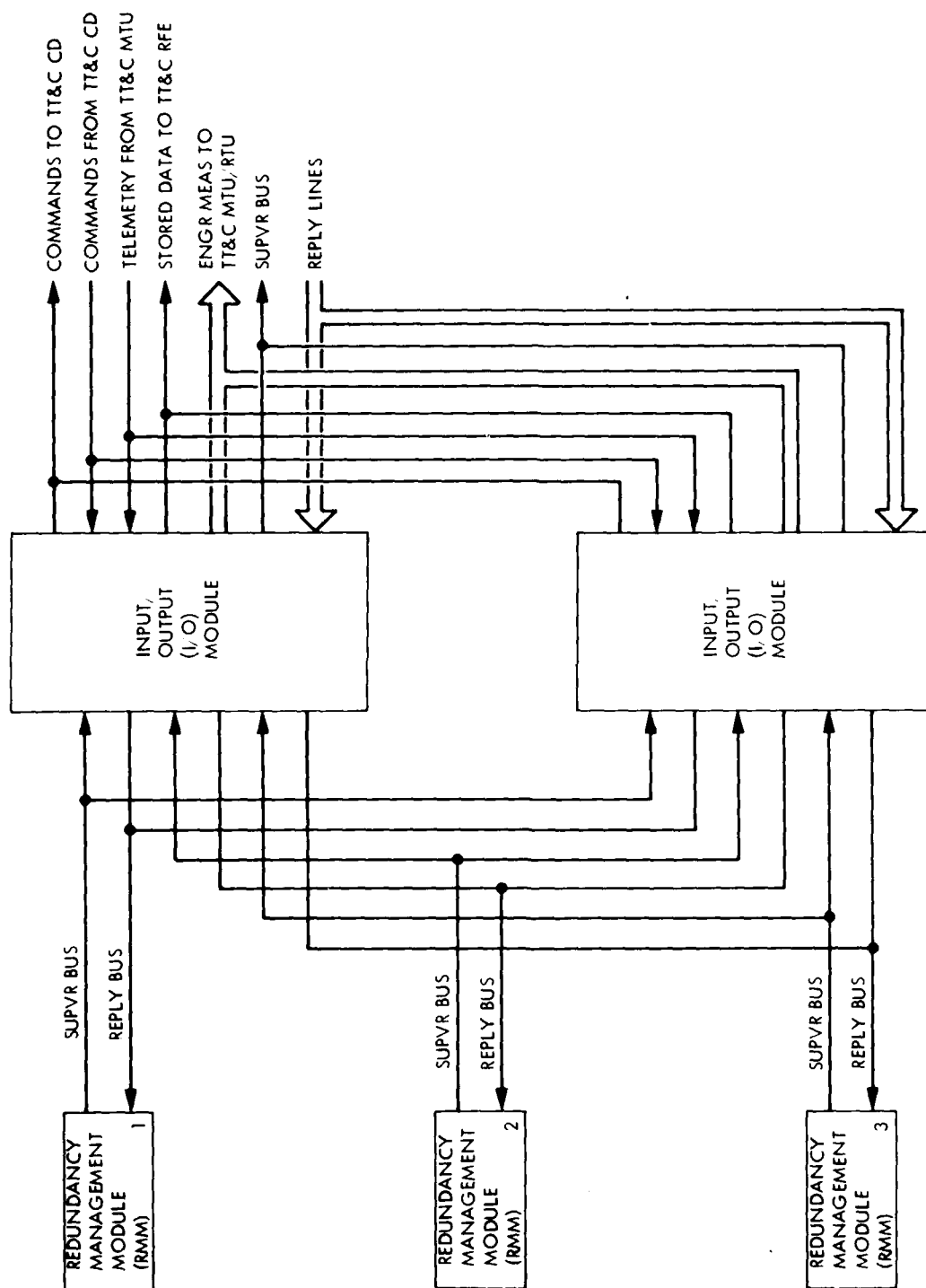


Figure 4-7. RMS Block Diagram

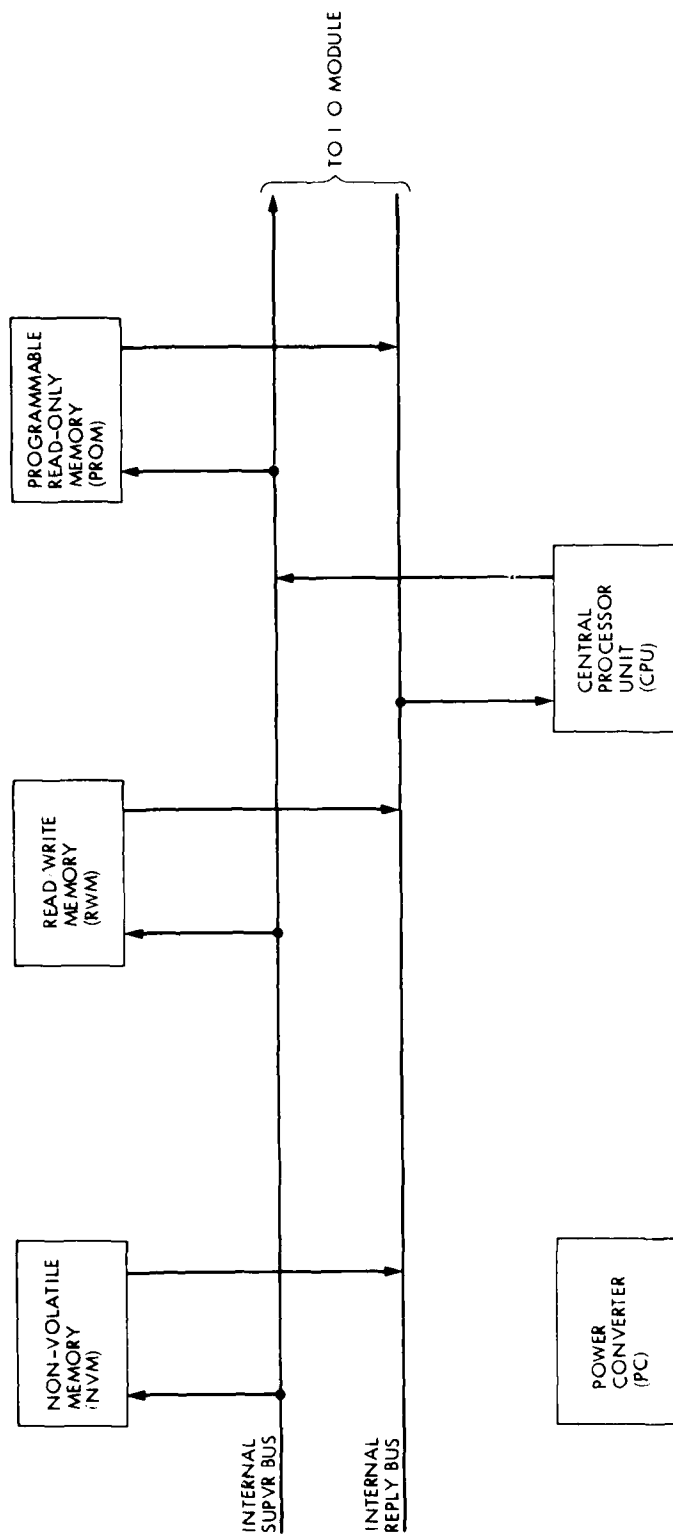


Figure 4-8. RMM Block Diagram

Referring to Figure 4-7, both I/O modules are individually powered and active. Any communications from an RMM CPU over a supervisory bus will normally be addressed to only one of the two I/O modules. Therefore, only the addressed I/O module will be able to transfer information to the RMS or TT&C subsystem. If a failure occurs in the selected I/O module, an RMM CPU detects this via several indications (lack of response from periodically issued diagnostic self-addressed commands to the TT&C CD, improper telemetry response to diagnostic data, etc.) and merely addresses further communications over the supervisory bus to the other I/O module.

A block diagram for an I/O module is given in Figure 4-9. As noted from Figure 4-9, commands addressed to a given I/O module will come from each RMM CPU over their respective supervisory busses to a majority voter unit. Therefore, fault correction and diagnostic commands to be issued externally through the command output unit will require that agreement of at least two of the three CPU outputs be achieved. If one RMM fails, RMS operation will be unimpaired since agreement will be attained from the remaining two RMM outputs. If failures occur in two or more RMMs, the majority vote agreement would not be achieved and the issuance of fault correction and diagnostic commands from the RMS would be inhibited unless RMS reconfiguration is commanded from the ground.

Referring to Figure 4-9, incoming commands from the TT&C CD can be routed to all RMMs simultaneously or any selected RMM via the designated reply busses. An application where a command would be addressed to a single RMM is a request from the ground to interrogate diagnostic data stored in the NVM of a given RMM. The CPU of the addressed RMM would execute readout of its NVM. The data would be routed over the RMM reply bus, on a noninterference basis, to the stored data output unit of the designated I/O module. Readout of the diagnostic data from the NVM of each remaining RMM could also be requested individually via properly addressed ground commands.

Since 1) reply line data from subsystem DPUs and 2) the telemetry stream data from the TT&C CD are available to all RMM CPUs via individual reply lines, this approach of individual readout of RMM diagnostic data could allow for multiple failures, i.e., two of the three RMMs could fail and valid fault detection, isolation, and required correction information could still be transferred to the ground by the good RMM. This not only would benefit mission operations activities on the ground related to fault detection and correction, it would also provide information as to which RMMs had failed. Furthermore, the design of the I/O module could be effected such that the majority voter unit could be bypassed with any selected supervisory bus via a single ground command. This would allow an undegraded fault correction capability to be reinstated with only one of the three RMMs operational. Since the selected RMM CPU could detect a failure in a given I/O module and subsequently address the remaining I/O module, multiple failures, including total failure of everything but one RMM and one I/O module, could theoretically allow undegraded spacecraft autonomy functions to still be achieved.

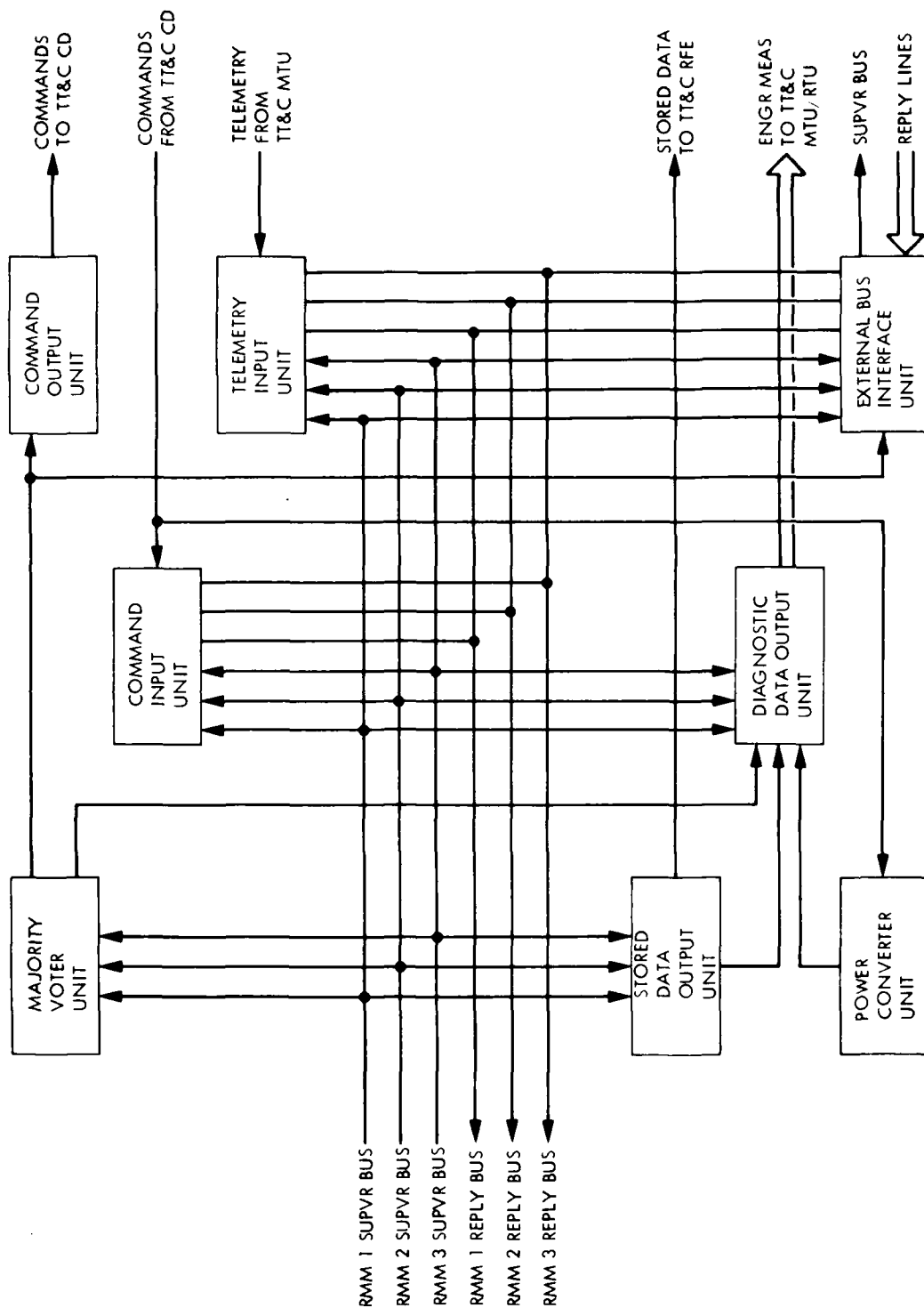


Figure 4-9. I/O Module Block Diagram

As noted in Figure 4-9, a discrete command could be sent directly from the TT&C CD to a power converter in the I/O module via a dedicated line to switch power on and off. This would allow complete disablement of one or both I/O modules via ground command. The net result would allow reversion of the DSCS III spacecraft to the nonautonomous condition with no degradatory effects with respect to current design and operational characteristics. This aspect of the defined architecture should greatly reduce the initial risk of integrating autonomy into the existing DSCS III flight-qualified system design.

4.1.2.5 DPU Design Description. A block diagram for a candidate DPU design is provided in Figure 4-10. A standard bus adapter circuit is provided for two-way communication with the RMS. Commands from the RMS are routed to all DPUs via the common bus. A DPU responds only to commands addressed to its host subsystem. As noted from Figure 4-6, pertinent subsystem-unique measurements are accessed by means of the subsystem I/O block. The measurements are in the form of analog and bi-level signals. The subsystem I/O block provides multiplexing, analog-to-digital conversion, and buffering functions. The memory stores subsystem-unique software subroutines for performing specific processing algorithms associated with anomaly detection and recovery procedures. Upon receipt of appropriate executive control commands from the RMS, the CPU executes these subroutines using incoming information from the subsystem I/O block. Processing results are stored in memory for interrogation by the RMS. The RMS can 1) load the DPU memory via the RMS supervisory bus and 2) readout the DPU memory over the DPU external reply line. The utilization of the separate reply lines for each DPU provides isolation between DPU responses to protect against the affects of anomalous operation by any given DPU.

The DPU interface with each subsystem is internally block redundant. The RMS provides near-continuous diagnostic health analysis of the active DPU blocks by sending diagnostic commands over the RMS supervisory bus and monitoring the reply line responses. If a DPU anomaly is detected, the RMS sends a discrete command via the TT&C subsystem to switch to a redundant DPU block.

4.1.2.6 Implementation Considerations. Candidate implementations of the RMS and DPU designs defined in Sections 4.1.2.4 and 4.1.2.5, commensurate with the design assumptions of Section 4.1.2.2, are described as follows:

4.1.2.6.1 Redundancy Management Module (RMM). The RMM of Figure 4-8 consists of five discrete functional blocks (CPU, PROM, RWM, NVM, and PC). The central processing unit (CPU) would be implemented using an 1802 microprocessor. The CPU peripheral direct memory access (DMA), timing, and control logic circuits would be implemented using CMOS 4000 series logic chips. Both the 1802 microprocessor and CMOS 4000 series logic chips have been radiation hardened and flight qualified on the JPL Galileo program.

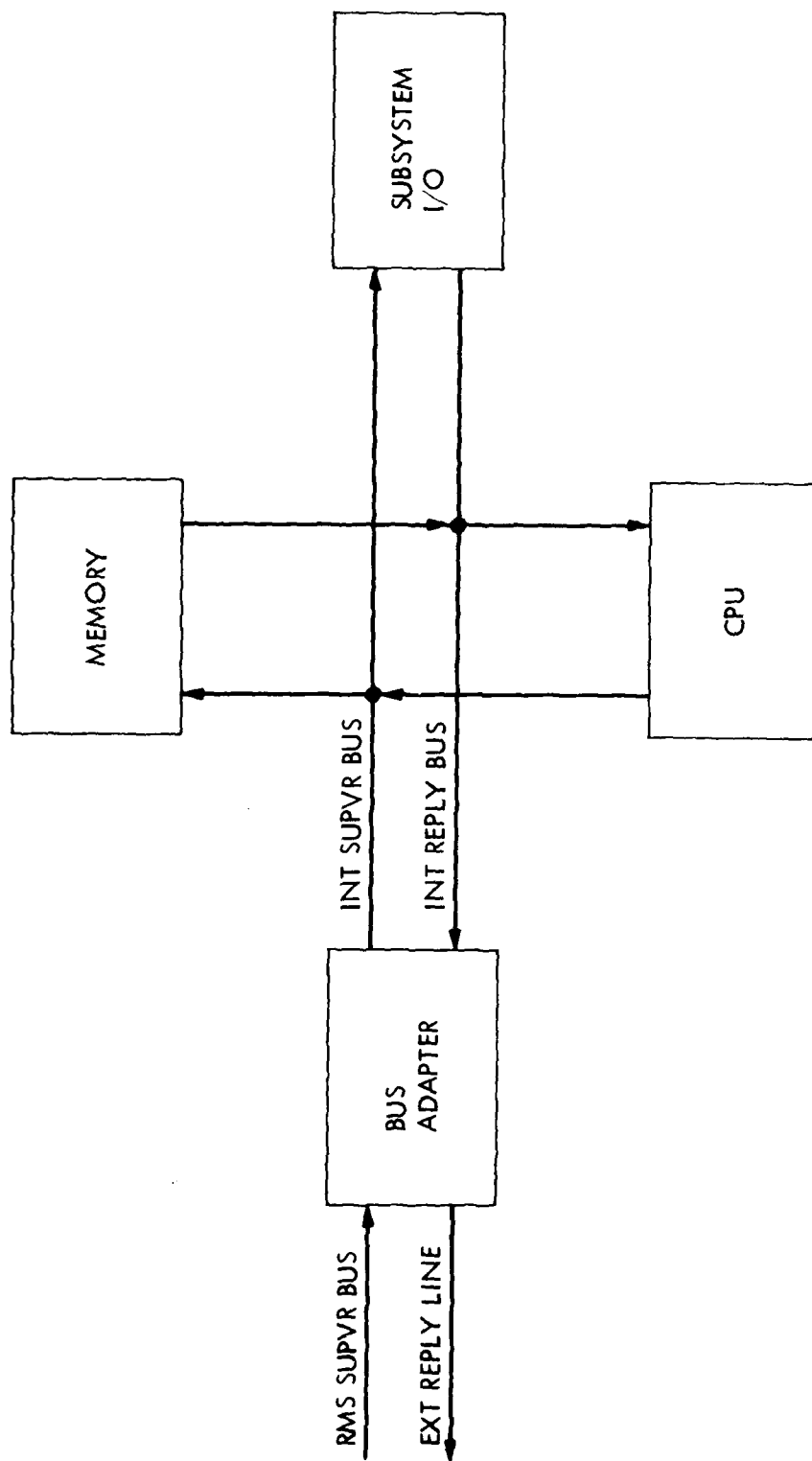


Figure 4-10. NPU Block Diagram

The programmable read-only memory (PROM) would be implemented using a commercially available, radiation-hardened, flight-qualified unit similar to that already included in the DSCS III TT&C MTU. The RMS PROM would be sized for 2 kilowords (8 bits/word) of memory which should be accommodated by a single chip.

The read-write memory (RWM) would be implemented using 256 x 4 bit TTC 244 CMOS memory chips which have also been radiation hardened and flight qualified on the JPL Galileo program. It would be sized for 8 kilowords (8 bits/word) of memory. This would require about 136 chips. Ninety-six of the 136 memory chips would be required for storage while the remaining 40 chips would be used for driver and decoding functions.

The nonvolatile memory (NVM) would be implemented using a commercially available, radiation-hardened 10⁶ bit bubble memory chip. The NMOS chips currently used in commercially developed bubble memory units (for memory control, timing, and driver circuits) would be replaced with radiation-hardened, flight-qualified CMOS and bi-polar chips.

The power converter (PC) would be implemented using radiation-hardened, flight-qualified components and circuit designs from the JPL Galileo program.

Packaging of the individual CPU, PROM, RWM, NVM, and PC functional blocks would be accomplished using Galileo leadless carrier packaging techniques. the resultant functional blocks would be housed and interconnected in a standard Galileo subchassis. Based upon the designated component types and memory sizing, the area provided by one side of a standard Galileo subchassis (41.3 cm x 18.4 cm) should be adequate to accommodate an entire RMM.

4.1.2.6.2 Input/Output (I/O) Module. The I/O module of Figure 4-9 consists of eight discrete functional blocks. All of these blocks would be implemented using radiation-hardened, flight-qualified components from the JPL Galileo program. The logic circuits and output buffers would use CMOS 4000 series logic chips. The power converter design and implementation would be identical to that used for an RMM.

As for an RMM, packaging of the individual functional blocks of Figure 4-9 would be accomplished using Galileo leadless carrier packaging techniques. The resultant functional blocks for both I/O modules would be housed and interconnected on one side of a standard Galileo subchassis.

4.1.2.6.3 Distributed Processing Unit (DPU). The candidate DPU design of Figure 4-10 consists of four functional blocks. The bus adapter circuit design would be very similar to that for the Galileo Command and Data Subsystem (CDS) bus adapter. The subsystem I/O block would use already designed circuits from the Galileo low-level module (LLM) design. The memory would use a 2 kiloword PROM and an 8 kiloword RWM identical to the designs used in the RMS. In like manner, the CPU would use an 1802 microprocessor identical to that used for the RMS.

It is assumed that a DPU would receive power from its host subsystem so that a dedicated power converter for a DPU is not required. It is further assumed that only radiation-hardened, flight-qualified components from the Galileo program would be used to implement a DPU.

4.1.2.6.4 Power, Weight, and Size. The estimated power, weight, and size requirements for the RMS defined herein are provided in Table 4-2. Referring to Table 4-2 each RMM of Figure 4-7 is estimated to require an average power of 2 watts for a total average power of 6 watts for three RMMs. The total peak power for all three RMMs should not exceed 12 watts. This could only occur when and if data were transferred to and from nonvolatile memories in all RMMs simultaneously. Each I/O module is estimated to require an average and peak power of 1 watt for a total power of 2 watts for the two modules of Figure 4-8. As noted from Table 4-2, the total estimated average power for the RMS is therefore 8 watts.

The weight of the RMS is based upon the use of two Galileo subchassis, each weighing 1 kg. The unit weight, with interconnection board, of each RMM is also estimated to be 1 kg. Since an RMM is mounted on one side of a Galileo subchassis, two subchassis are required to house the three RMMs of Figure 4-7. The three RMMs therefore add 3 kg to the 2 kg required by the two subchassis housings. The two I/O modules of Figure 4-7, each weighing 0.5 kg (including interconnection boards), are mounted on the remaining empty side of one of the subchassis. This results in a total estimated RMS weight of 6 kg.

The two Galileo subchassis used to house the RMS each measure 41.3 cm x 18.4 cm x 4.5 cm for a per unit volume of 3420 cm³. The total estimated volume required by the RMS is therefore 6840 cm³.

The estimated power, weight, and size requirements for the DPU defined herein are provided in Table 4-3. Since redundant DPU blocks will not be powered, the total average power imposed upon a host subsystem is the power required by one non-redundant DPU. This is estimated to be 2 watts. Since two non-redundant DPUs (each weighing 1 kg) are housed in a single Galileo subchassis (which also weighs 1 kg), the total weight estimate for a redundant DPU with a subchassis housing is 3 kg. The volume of 3420 cm³ is based on the external dimensions of the standard Galileo subchassis housing.

4.1.3 Data Memory Subsystem (DMS)

Expansion of the RMS capability defined in Section 4.1.1 to the "extensive" RMS system design architecture described in Section 4.1.2 dictates the need for a spacecraft nonvolatile mass storage capability significantly exceeding that provided by the RMS alone. Section 4.1.3 describes a candidate design approach for a Data Memory Subsystem (DMS) to accommodate this need.

Table 4-2. RMS Implementation Characteristics

ELEMENT	UNIT POWER watts	UNIT WEIGHT kg	UNIT VOLUME cm ³	QUANTITY	TOTAL POWER watts	TOTAL WEIGHT kg	TOTAL VOLUME cm ³
RMM	2	1	—	3	6	3	—
I/O	1	0.5	—	2	2	1	—
SUBCHASSIS	—	1	3420	2	—	2	6840
TOTALS	—	—	—	—	8	6	6840

Table 4-3. DPU Implementation Characteristics

ELEMENT	UNIT POWER watts	UNIT WEIGHT kg	UNIT VOLUME cm ³	QUANTITY	TOTAL POWER watts	TOTAL WEIGHT kg	TOTAL VOLUME cm ³
DPU	2	1	-	2	2	2	-
SUBCHASSIS	-	1	3420	1	-	1	3420
TOTALS	-	-	-	-	2	3	3420

4.1.3.1 Requirements. Autonomous operation of the DSCS III spacecraft will require periods up to 6 months without ground intervention. Assuming that a compression ratio of two orders of magnitude can be achieved from the DSCS III telemetry stream, storage of telemetry status information at an average rate of 10 bps would require a storage capacity of almost 2×10^8 bits during a period of 6 months. In addition, significant 1) analysis and corrective action response data associated with detected faults, 2) resource management data, and 3) software programs for the entire spacecraft must be stored by the DMS throughout the period of autonomous operation. Therefore, a nonvolatile storage capability of between 1×10^8 and 1×10^9 bits is considered necessary for the DMS.

4.1.3.2 Design Description. DMS interface characteristics, applicable to the "extensive" RMS system design architecture, are shown in Figure 4-6. Referring to Figure 4-6, it is noted that all spacecraft requested data transfers (diagnostics, software, etc.) are routed to and from a DMS DPU via the RMS supervisory bus under executive control of the RMS. Through the DPU interface, the RMS controls the DMS operations and performs health diagnostic functions for the DMS. The RMS nonvolatile memory described in Section 4.1.1 would function as a buffer for the DMS. When filled, the contents, or a portion thereof, of a 1×10^6 bit RMS bubble memory would be transferred to the DMS by the RMS. Also, the RMS would be responsible for reloading all volatile memories of the spacecraft with software programs from the DMS as required. An exception to RMS control of data transfer to and from the DMS would be in the case of ground intervention. The ground would be able to interrogate the DMS directly through the TT&C subsystem, independent of the RMS.

For a DMS storage requirement exceeding 1×10^8 bits, the only candidate technology that is feasible for a DSCS III application in the near term is magnetic tape. Current bubble technology based on the use of 1×10^6 bit bubble memory units (as proposed for the candidate RMS design of sections 4.1.1 and 4.1.2) starts becoming inferior to magnetic tape, from a cost, power, and weight standpoint, for capacities exceeding 1×10^7 bits. A candidate DMS design approach for an autonomous DSCS III spacecraft assumes use of two, redundant, flight-qualified, radiation-hardened, Galileo, digital tape recorders (DTRs).

Design and performance characteristics for the candidate DMS design are summarized in Table 4-4. Referring to Table 4-4, each DTR provides a capacity of 9×10^8 bits so that the total possible capacity, if both redundant units were used, would be 1.8×10^9 bits. The life of a DTR is determined by the number of start/stop cycles and tape passes it must perform. The specified capability values for the Galileo DTR, shown in Table 4-4, far exceed what would be accumulated on a DSCS III-type mission over a 10-year period. Therefore, the proposed DTR design approach should be highly reliable.

Table 4-4. DMS Performance Capability Summary

PARAMETER	CAPABILITY
STORAGE CAPACITY	9×10^8 bits/DTR
ERROR RATE	< 5 in 10^6 bits
START/STOP CYCLES	$\geq 20,000$ /DTR
TAPE PASSES	$\geq 15,000$ /DTR
DATA INPUT RATES	10 kbps TO 160 kbps
DATA OUTPUT RATES	10 kbps TO 160 kbps
RADIATION	$\geq 150,000$ rads

4.1.3.3 Implementation Considerations. The candidate DMS implementation consists of two DTRs. Each DTR would be packaged as a separate, hermetically sealed unit. It would be identical to a Galileo DTR with the exception of specific input and output data rate requirements. Therefore, each DTR, as a complete entity, would already be flight qualified and radiation hardened to Galileo requirements.

The power, weight, and size characteristics of the proposed DMS are summarized in Table 4-5. Since only one DTR would be powered at any time, the average power would be due to one DTR. The duty cycle for the record and playback modes is so low that the standby power of 3 watts represents, for all practical purposes, the average power. The weight and volume of 18 kg and 28,454 cm³ for a DMS containing redundant DTRs is based on actual mass and dimensional characteristics of the Galileo DMS.

4.1.4 A Redundancy Management Subsystem Application Example

As an exercise in estimating word requirements for various anomaly response algorithms, one fairly well documented algorithm from the DSCS III Orbit Operations Handbook was selected for analysis. This was the battery high temperature recovery procedure of Section 6.3.1.1 of the Handbook. The procedure is described by Figure 4-11. This procedure requires, as an input, battery temperature and an indication of which battery that temperature is associated with. This data is available within the current telemetry data or can be implied from it (which battery) with no requirements for additional measurements.

Normal battery temperature is specified as 0° C to 20°C. The high temperature recovery algorithm is activated whenever the battery temperature exceeds 20°C. It attempts to correct the temperature anomaly primarily by controlling the battery charge profile if the temperature is below 32°C. Above 32°C the charger is turned off and other areas such as a battery heater failure are investigated. The heater anomaly response is not included in this study, however.

4.1.4.1 Assumptions. In attempting a rudimentary assessment of this type, certain assumptions were made to simplify the effort. These assumptions were:

- (1) An RMS architecture as described in Section 4.1.1.
- (2) A 1.6 Mhz clock.
- (3) An 1802 microprocessor powered at 10 volts.
- (4) The existence of a software telemetry decommutation routine that calls response algorithms, such as the one discussed in this section, each time the associated telemetry measurement is decommutated.

Table 4-5. DMS Implementation Characteristics

AVG POWER watts	MAX POWER watts	WEIGHT kg	VOLUME cm ³
3	20	18	28,454

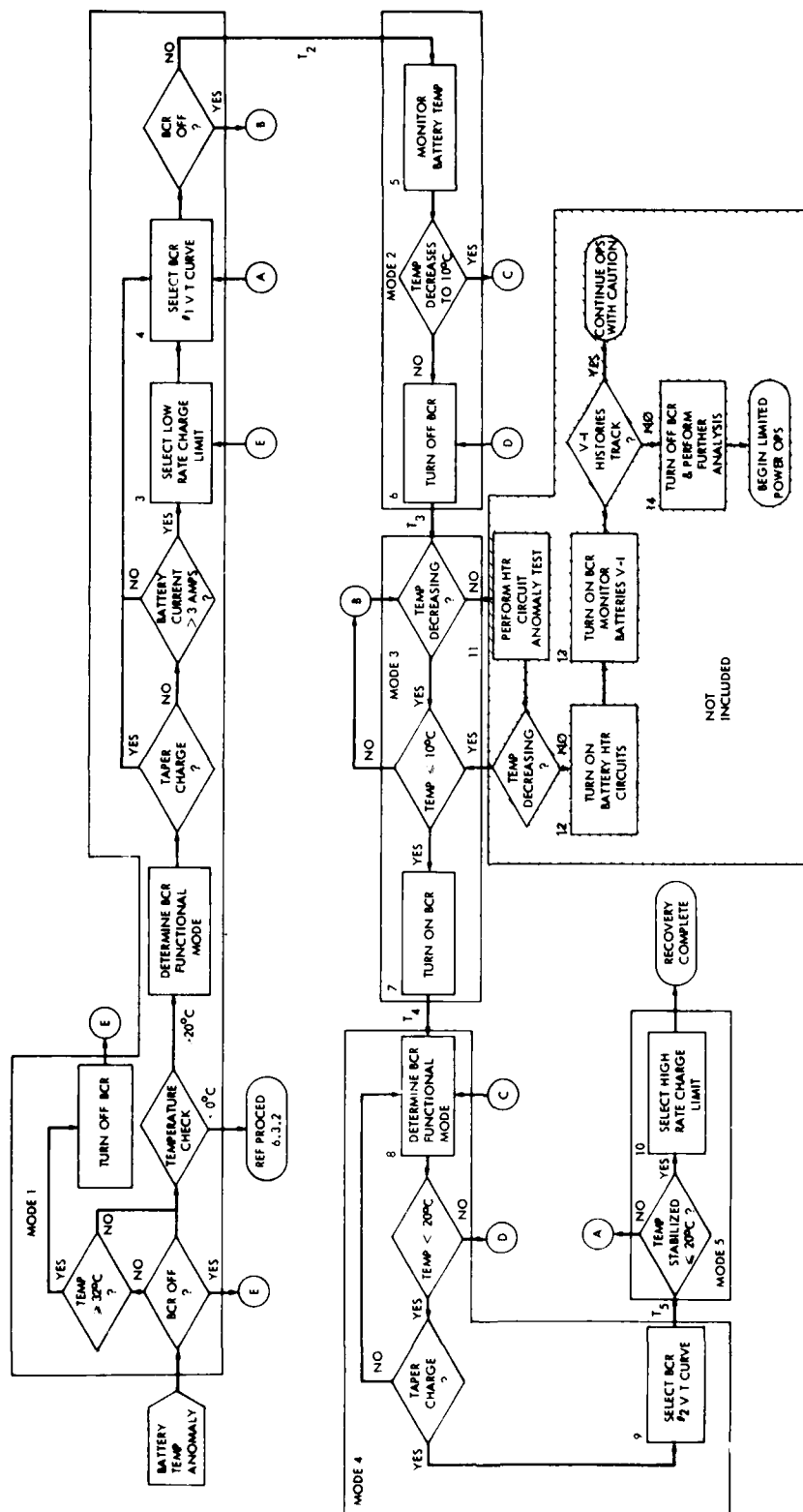


Figure 4-11. Battery HI-Temp Anomaly Contingency Events

- (5) The existence of an output routine that will output a designated command to the TT&C command decoder.

The required recovery response is summarized in Figure 4-11.

4.1.4.2 Anomaly Response Description. In general, anomaly responses typically require some initial action followed by a time delay to analyze the effect of the initial action, then an additional action depending on the effect of the previous action, and so on. In other words, act, wait (analyze); act, wait (analyze), act, wait (analyze); etc.

The anomaly response studied was a prime example of that structure. Figure 4-11 (from the DSCS Operations Handbook) shows the actions (called Modes) followed by the wait period for the next action (T_m).

It was assumed that a telemetry decommutation routine (not assessed) provides the current value of the battery temperature and the battery number to the anomaly response routine each time the measurement appears in the telemetry stream. The battery number is required because there are three batteries on board, each of which could be at a different temperature, or following a different charge profile.

The battery charge profile typically follows one of 4 voltage/temperature curves which must be modeled by the high-temperature response routine. Fortunately, the curves can be simply modeled by a 16 word look-up table and a fixed bias increment between one curve and the next.

The battery high temperature response was coded in 1802 assembly language without any great effort to optimize (minimize) the number of words required, or the basic software structure assumed.

4.1.4.3 Design Results

The word estimate for this response is as follows:

Mode 1	209 words
Mode 2	25
Mode 3	27
Mode 4	41
Mode 5	29
Variables	14
V/T curve approximation	16
TOTAL	361 words

The count for mode 1 is significantly higher than the other modes because there is more analysis required. Also, several subroutines are included which are used by more than one mode.

AD-A106 064

JET PROPULSION LAB PASADENA CA
ASSESSMENT OF AUTONOMOUS OPTIONS FOR THE OSCS III SATELLITE SYS--ETC(U)
AUG 81 D L PIVIROTTI, M MARCUCCI

F/G 22/2

NAS7-100

UNCLASSIFIED

JPL-7030-2-VOL-3

SD-TR-81-87-VOL-3

NL

2 OF 3
40 A
106 URA

END
DATE
FILMED
11-81
DTIC

The longest path (mode 1) requires approximately 430 machine cycles, which at the assumed 1.6 Mhz clock frequently would require slightly over 2.1 milliseconds to execute (1 machine cycle equals 8 clock cycles).

Although the 2.1 milliseconds required for the longest path is well within the 8 millisecond spacing of telemetry channels in a 2.048 second mainframe, processing time required by the assumed decommutation and output routines may dictate that telemetry channels requiring additional processing not be contiguous in the mainframe. This possible constraint would not appear to be of any concern, however, since telemetry positions are typically somewhat arbitrary. (This may not be true of DSCS III, however.)

In general, this routine is fairly simple and not particularly demanding of the particular architecture assumed. The fairly large number of words required, however, portends the need for a significant amount of memory when all routines for DSCS III autonomy are considered.

4.2 AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF POWER FUNCTION*

Figure 4-12 presents a functional hierarchy of power function integrity maintenance. Autonomous options are presented in the following paragraphs to protect the power system from user overloads and to protect the spacecraft against loss of power due to battery chain or power converter failures.

4.2.1 Isolation/Protection from User Load Anomalies

4.2.1.1 Isolate Load Faults - Level 3/Category I. This function is automatic through the use of fuses.

4.2.1.2 Protect Bus from User Overloads - Level I/Category I. In order to implement an automated on-board capability to select alternate (redundant) loads it appears that a large increase in the number of diagnostic measurements would be required. Strictly speaking, the selection of alternate loads is not a power function, but must be performed by equipment associated with the appropriate subsystems, particularly if health tests are required. Given a decision that an alternate load is needed, the power subsystem could then provide the necessary switching.

Needed are:

- (1) Measurement of all load voltages and currents
- (2) Power profile for all loads
- (3) Processing to compare actuals with predicted values
- (4) Load management capability

4.2.2 Reconfiguration Capability After Internal Faults

4.2.2.1 Protect Against Battery Chain Failure - Level 2/Category I. An automated on board capability to autonomously work around battery chain failures will require measured battery parameters, a stored battery state of charge model and stored solar array and load profile predictions. There may be a need to flatten peak load demands during the day through the use of load (including thermal) management to ensure adequate battery charge. Two battery chain failures will require eclipse load management to maintain thermal requirements and possibly make allowance for some communications loads. If all batteries fail, the spacecraft will have to perform load management to keep all peak loads below the array capability and utilize a spacecraft shutdown procedure for eclipses.

*By R. C. Detwiler, T. W. Koerner and G. W. Wester

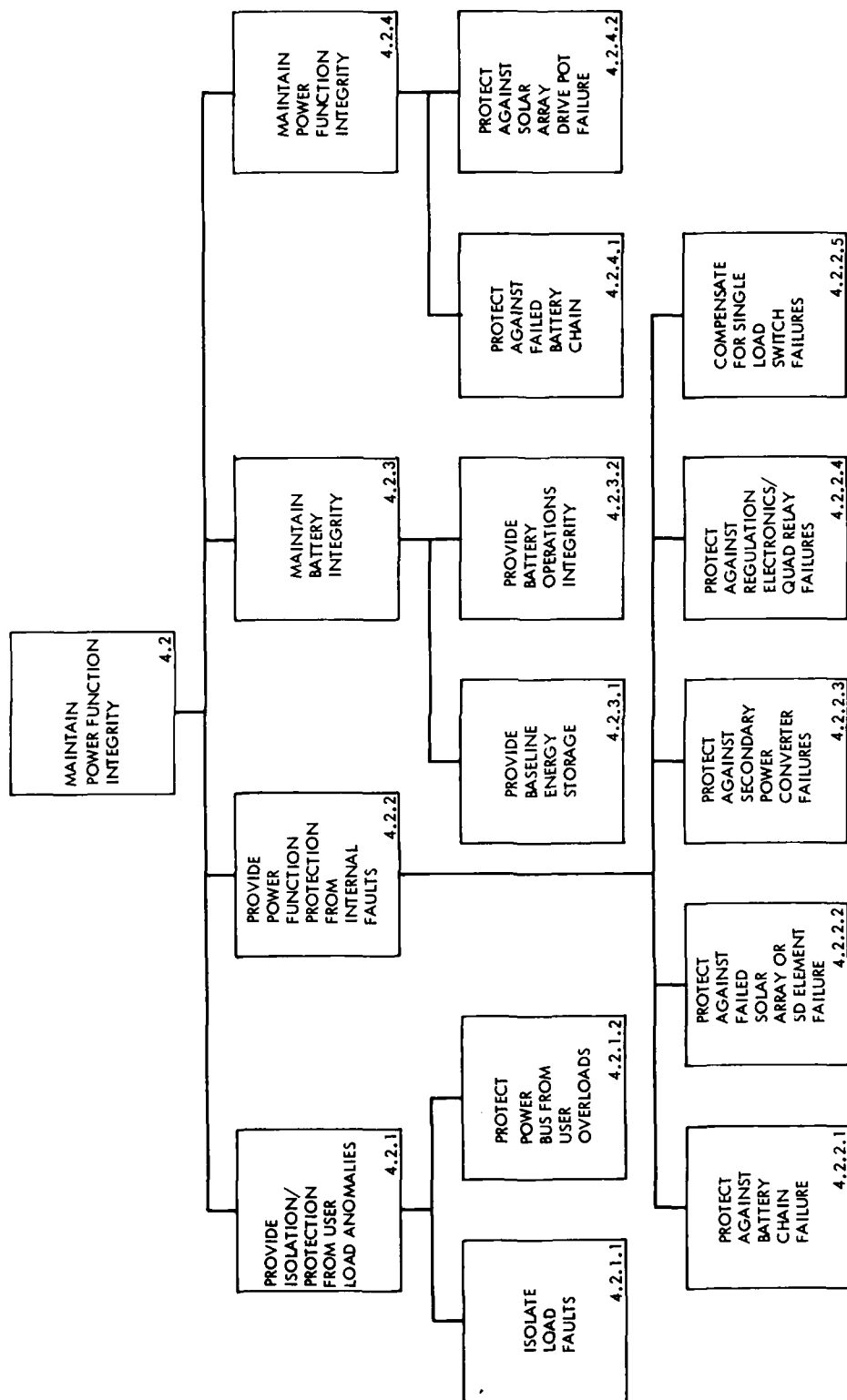


Figure 4-12. S/C Power Integrity Maintenance Functional Hierarchy

Additional Capability Required:

- (1) Battery state of charge model
- (2) Battery charge history
- (3) Load profile predictions
- (4) Array profile predictions
- (5) Load management capability
- (6) Thermal load management
- (7) S/C shutdown and startup program

4.2.2.2 SA or Shunt Dissipator Element Failure - Level 3/Category I.
Action taken (Load management or control of battery charging parameters)
should be the same as for management of battery state of charge.

4.2.2.3 Secondary Power Converter Failure - Level 2/Category I. If a
converter failure is suspected, both input and output voltage measurements
would be required to distinguish this case from a source or load fault.
Multiple converter outputs make this difficult as does the automatic selection
task itself.

Needed are:

- (1) Measurement of all converter output voltages and currents
- (2) Measurement of all converter input voltages and currents
- (3) Processing to analyze problems
- (4) Drive circuits to operate converter selection relays

4.2.2.4 Regulator Electronics/Quad Relay Failures - Level 5/Category I.
These elements are redundant and there are no changes required for autonomy.

4.2.2.5 Load Switch Failures - Level 1/Category I. For complete
autonomy without the benefit of ground detection of load operating
performance, detection of load switch failures must be inferred from telemetry
or internal power measurements in comparison with expected load power changes
when a load is added or removed. Once a load switch failure is identified the
load profile must be modified to account for the failed switch.

This option requires the following:

- (1) Stored table of nominal power for all modes of each load
- (2) Measure of the power change each time a load is switched
- (3) Comparison of measured and expected power change
- (4) Identification of load switch failures and their state
- (5) Modification of load profile.

4.2.3 Maintain Battery Integrity

4.2.3.1 Provide Baseline Storage Capacity - Level 1/Category II. It is unlikely that the batteries in the DSCS spacecraft would need reconditioning for several years, since only about 85 discharge cycles are encountered each year, and not all of these are deep discharges. It is questionable whether it is necessary to incorporate automatic battery reconditioning for this type of mission. A simpler approach might be to start the reconditioning cycle by ground command early in a period of no eclipses and have the spacecraft automatically complete the cycle and restore the battery to normal use. This would ensure normal operation during eclipses in the event that ground command capability was lost. On the other hand, if the on-board system is sophisticated enough to handle the problem of accurate battery charge status control, it is probably a small additional change to include on-board, autonomous reconditioning.

Additional capability required:

- (1) Simplified Approach
 - (a) Battery voltage discharge limit sensor
 - (b) Relay drive circuits to restore battery to normal operation
- (2) Fully Automated Approach
 - (a) Battery state of charge model
 - (b) Battery charge history
 - (c) Battery capacity model
 - (d) Eclipse season prediction
 - (e) Processing for above

4.2.3.2 Provide Battery Operations Integrity - Level 3/Category I.

4.2.3.2.1 Control Battery Temperature. The battery over temperature sensor is in the current DSCS III hardware so that automation of this function only requires some additional software to be added to the computing capabilities outlined in Para. 4.2.2.1 above.

4.2.3.2.2 Control Battery Discharge Time. The battery discharge timer function would be eliminated and replaced by the software/hardware used to manage stored energy (see Para. 3.1.2 above). Instead of a timing function the true Depth of Discharge (DOD) of the battery would be calculated on a continuous basis. Navigation information in the form of array output predictions would provide the computing function with knowledge of the length of each eclipse as well as double eclipse events so that load management, if required, could be planned. In the event that DOD limits were still approached, the computing function would take the drastic measures required for spacecraft survival.

4.2.3.2.3 Control Battery Survival Mode. Loss of earth lock sensing by ACS would be fed into the power load management function where loads not essential to the earth acquisition function would be removed from the power bus until earth lock was reestablished.

4.2.4 Maintain Power Function Integrity

4.2.4.1 Protect Against Failed Battery Chain - Level 2/Category I. Battery chain failures are monitored by the computing function which in turn performs load management based on the extent of the failure. Once all batteries have failed the lack of accurate navigation information, due to the loss of an accurate clock, will require a load shedding routine once a bus undervoltage is sensed. Those loads sensitive to reduced input voltages would be dropped from the bus via a redundant power switch powered by capacitor storage. Upon exit from eclipse the power management function can begin turning on loads in an orderly fashion to prevent instability of the power bus.

4.2.4.2 Protect Against Solar Array Drive Potentiometer Failure - Level 2/Category I. Autonomous fault detection of the SAD pot function, and correction by switching to the redundant pot can be accomplished by the moderately redesigned ACS computer/electronics. Software fault detection logic can utilize information already available such as the SAD stepper motor commanded pulse rate, SAD port A/B powered states, pot A or B selection, pot output signals, and software timer.

The commanded SAD pulse rate (normal orbit rate) is preprogrammed and keyed to a multiple of ACS CPU cycles. A health check of the pot readout can be made by comparison with the SAD command. An incremental change in pot angle should equate to an average of several pulse train increments. Every 16 CPU cycles the stepper motor is given 22 pulses. Thus in approximately 16 seconds a known incremental angle is driven, and the pot reading should correspond within instrumentation tolerances and resolution. If agreement is not present then the comparison can be extended to the redundant pot for positive separation of a pot fault from a stepper motor or drive fault. Since both SAD's drive a common shaft to rotate the North and South solar panels, the wipers of both pots must be moving together, and only excitation to a pot is required to obtain a readout signal. This can be autonomously done by the ACS modified for access to its own power relay matrices.

4.3 AUTONOMOUS OPTIONS TO MAINTAIN S/C ATTITUDE CONTROL FUNCTION*

Figure 4-13 illustrates the hierarchy of the attitude control maintenance function.

4.3.1 Perform Routine Health Checks and Maintenance - Level 2 to 4/Category I

Autonomy of this health check analysis function should be implemented in the larger context of fault tolerant design. The data acquisition and transfer and command protection functions which are already autonomous provide a part of the overall fault tolerant system. Routine health monitoring is only the beginning of the fault tolerant design process, and is totally contingent on the autonomous use of the sensed data, i.e., data analysis, fault identification, fault isolation, decisions for fault correction, and enablement of alternative functions or elements. Thus, significant design changes are required to add this capability for an autonomous ACS.

4.3.2 Configure S/C for External Events

4.3.2.1 Maintain Attitude During Eclipse - Level 2/Category I. Sun/lunar eclipse compensation can be made autonomous contingent on an autonomous navigation function since knowledge of orbit state and earth-sun-moon relationships is needed to set up the event timeline and prevent false or ambiguous interpretation of reduced sun sensor signal levels.

4.3.2.2 Protect Against Loss of Earth Presence Signal - Level 3/Category I. By providing an autonomous earth reacquisition function in lieu of the survival mode currently directed, the autonomous earth acquisition/reacquisition sequence (see 2.2.1.4) could be initiated.

4.3.2.3 Recover From a Nuclear Event - Level 5/Category I. Recovery should include autonomous reference reacquisition as described in Section 2.2.1.4.

4.3.2.4 Protect Earth Sensor Against Sun - Level 5/Category I.

4.3.2.5 Yaw Rate Reduction Gyro Backup - Level 2/Category I. This can be automated with the addition of S/W for fault sensing/correction or as a normal protection mode for specific events.

*By D. J. Eisenman, (G.E.), J. R. Matijevic and E. Mettler

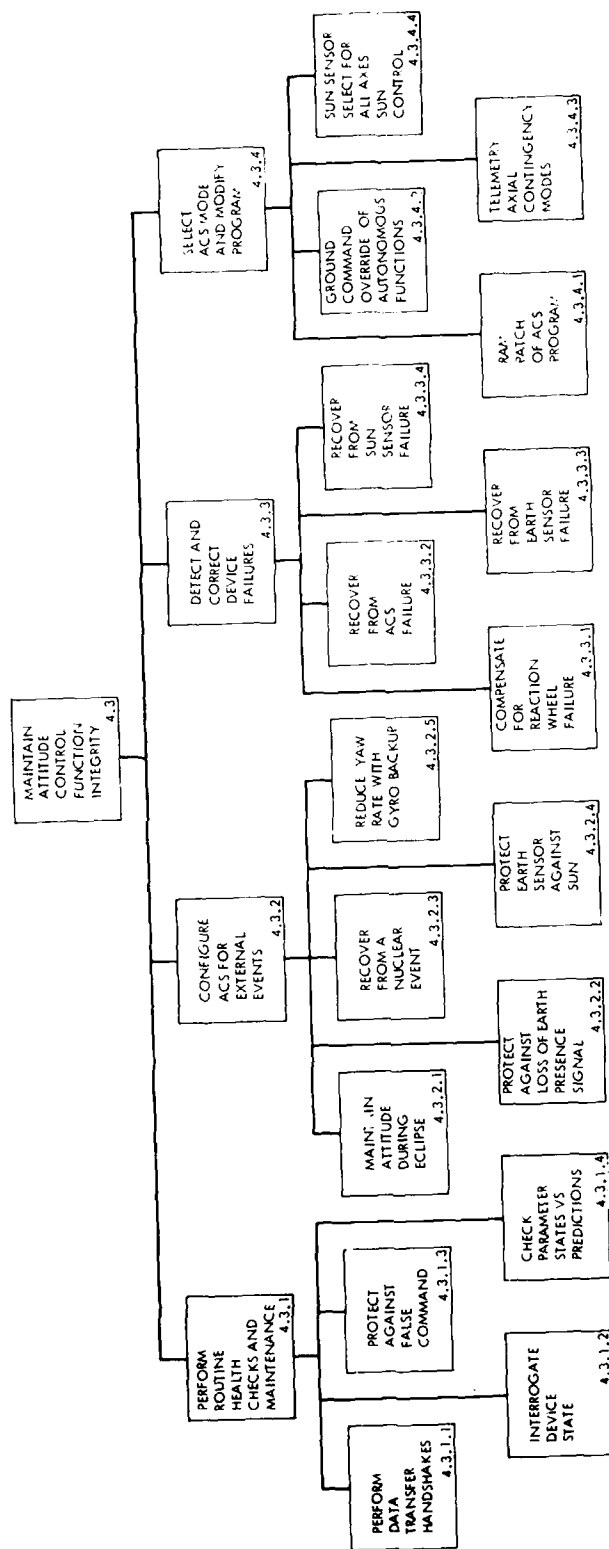


Figure 4-13. Attitude Control Integrity Maintenance Functional Hierarchy

4.3.3 Detect and Correct Device Failures - Level 2/Category I

Device(s) failure detection/correction, i.e., reaction wheels, Attitude Control Electronics (ACE), and sensors will require added capability to automate what is now done by ground controllers. This includes analysis of telemetry information and uplink of commands for spares switching. To do this on board autonomously requires large blocks of new S/W which is likely to exceed the present capacity of the ACS computer. In addition, the determination of an ACS computer failure implies another active system processor monitoring the on-line ACE, and controlling the ACE spares switching. Thus significant software and computer hardware additions are required to implement general integrity maintenance of the ACS.

Selection of redundant Earth and Sun Sensor channels can be automated by addition of software to perform fault sensing, analysis, and channel enable/disable decisions.

4.3.4 Select ACS Modes and Modify Programs

ACS software program modification/reprogramming, alternative control mode selections, override functions, TLM format/sequence/frame rate changes, and contingency modes are also part of the previous total concept of fault tolerant, autonomous design, and are included in the need for major additions to software and processor hardware capabilities. Most of these functions are, by definition, ground commands, but can be made partially autonomous to improve the ease of ground command/execution on board.

4.3.4.1 RAM Patch of ACE Program - Level 2/Category III. By definition this is not an autonomous procedure.

4.3.4.2 Ground Command Override of Autonomous Function - Level 3/Category III. By definition this is not an autonomous procedure.

4.3.4.3 Telemetry "Axial" Contingency Modes - Level 2/Category I. This function could be automated as a part of the overall fault detection and correction function.

4.3.4.4 Sun Sensor Select for All Axes Sun Control - Level 2/Category II This should be a part of overall contingency planning/health checking/analysis.

4.3.5 Implementation Considerations for Autonomous Maintenance of the Attitude Control Function

4.3.5.1 Computer Resource Considerations for ACS Integrity Maintenance.

The existing ACE microcomputer is described in Appendix C of Volume II. Options for modest design changes for providing additional autonomy for the service functions are described in Section 2.2.4. This section addresses architectural design changes which might be required to perform the autonomous integrity maintenance functions outlined above.

4.3.5.1.1 Input Requirements. Bi-level indicators and analog signals currently within the ACS would have to be presented to the microcomputer to allow them to be processed and to verify the processing response. Only one user-configurable, external interrupt to the CPU is available within LS111 type systems. It cannot be determined if this interrupt line is being used in the ACE microcomputer design. The addition of similar special purpose interrupt lines would result in extensive modifications to the digital bus, CPU interface and require program code for each added interrupt. LS111 type systems have a flexible I/O module or port expansion capability. Special purpose interrupt lines could be 'simulated' by using I/O device vector interrupts. Special purpose I/O devices could be configured on the bus with the appropriate proximity to the ROM program. Operating in this manner appears contrary to the current device polling mode. It seems reasonable to assume that only a certain amount of ROM program timing disruption can be tolerated for device I/O vector operation just as only a certain amount of timing disruption can be tolerated for RAM patching. Direct interrupt lines cannot be used for analog signals.

The most attractive approach to provide new input would be to add a port or ports which function and are operated in a similar fashion to the existing sensor port. This port(s) would process both analog and bi-level digital signals. The CPU would then poll the device to acquire input for processing. Adding another port would require additional power of an undetermined amount. Presumably, any added port would be redundant. It would be necessary to add to the TT&C DC drive circuitry and the ACS latching relay matrix to select either port. Many spare TT&C DC drive circuits are available. The number of spare ACS latching relays available is not known. Program space would also be required to operate the port.

4.3.5.1.2 Output Requirements. Access to both ACS and EPDS (for earth sensor power switching, etc.) would have to be provided to manage ACS redundant blocks. Adding a new port to drive the input to the ACE resident DC relay matrix would provide complete command access to ACS. This function would drive the matrix as the TT&C CD now does but in a shared mode with TT&C. The interface may require changes to TT&C to provide DC driver circuit isolation. This port or a similar port would be used to issue EPDS DC commands. One port would be required for each DC relay matrix. To assure command access capability, both ports would remain active just as both of the existing DC relay matrices are now. These ports would require additional power of an undetermined amount. Program space would also be required to operate the port.

Alternatively, a port interface with TT&C would funnel ACS and EPDS DC commands from the microcomputer through the TT&C DC. This would be a complex port of a functional design similar to the telemetry port. The port would be required to format multiple microcomputer words into a single DC command. The port would be shared by TT&C and the microcomputer, requiring port access protocols for both. It would also need to be able to buffer commands for the times when the CPU is powered off and TT&C is busy processing commands from the ground.

Assuming the use of redundant ports, both would remain active to assure command access. The ports would require additional power of an undetermined amount and program space to operate them.

4.3.5.1.3 CPU/ROM Options. As outlined in Section 1.3.2.2, assuming that the redundancy management and health maintenance functions could execute in the absence of a ground command request for the BFN function, much time is available in each CPU execution cycle. However, if this assumption cannot be made, another reprogramming mechanism can be employed to execute autonomous functions during the remaining available time. Different autonomous functions would execute through successive CPU execute cycles and then repeat in whatever order and CPU execution cycle frequency is desired. This method would use the existing power strobe operation and thus provide whatever power savings which can be achieved.

Alternatively, a continuous background function could be implemented by changing the 1.024 power strobe interrupt into a clock strobe interrupt which would be used to restart the ACS/BFN program. This would leave the CPU/ROM always powered thus allowing a background function to use any time not consumed by the normal ACS/BFN program. Constant power to the CPU/ROM would increase the average power usage by about 60% (9.5 watts).

If the additional processing cannot be achieved as a background function to the BFN program, or in the remaining 12% of the existing CPU time available each cycle, the alternatives would be to increase the CPU clock speed or replace the CPU with a faster one to do more computing in each execution cycle. The feasibility of these approaches cannot be assessed with the information at hand. One could say, however, that these approaches could impact everything tied to the digital bus and the bus itself.

4.3.5.1.4 ROM Options. LS111 type microcomputers of this vintage can be expanded to 32K of memory space. However, 0-28K is recommended for memory address locations and 28K -32K for peripheral I/O device addresses. In fact, ACE microcomputer port addressing begins at the start of this upper 4K boundary. Included within this I/O device address boundary are the HRAM addresses. The first 8K of address space is taken up by the ROM. The next 2K is taken up by the prime and redundant RAM. This leaves 18K of memory address space into which additional ROM and RAM could be inserted. There is no information at hand to indicate that the ACE microcomputer could not be expanded to add more memory. The only restrictions appear to be power, weight

and the changes to the TT&C driver circuits and the ACS latching relay matrix required to provide for prime/redundant memory selection by discrete command. Discrete commands would also be required to provide an enable/disable selection for those autonomous functions stored in ROM.

4.3.5.2 Autonomous Attitude Control Function Architecture. The specific architecture of the attitude control function must be considered a part of the autonomous design. For instance the classic block-redundant approach, which satisfies the required reliability by powering-up and switching in dormant spares after entering a safe-hold mode, may not provide a proper solution for transparent validation. To provide for undisturbed payload services such as antenna articulation, synthetic array beamforming/pointing, stationkeeping, etc. certain approaches seem to offer advantages by virtue of embedded partitioning in the basic design.

4.3.5.2.1 Internal Device Redundancy. Internal device redundancy instead of external (block) redundancy is advantageous. This applies to all devices: sensors, computers, actuator motor windings, drive circuits and control logic. This is not a new technique, and in fact now exists in flight hardware in specific places in DSCS III i.e., the multi-detector earth sensor. The application of this powerful approach to the entire attitude control function is the idea which has been gaining preference in fault tolerant aircraft flight control systems. Internal partitioning of computer functions provides dual processing, memory, and monitor capabilities that greatly increase the fault handling capacity and effectiveness to screen out faults before they propagate through the system. By this approach to redundancy within a single computer the comparisons between redundant computations and sensed data are continuously monitored by fault detection algorithms. A second, fully redundant computer with internally dual structure acts as the spare unit.

4.3.5.2.2 Partitioning/Cross Strapping. Deep partitioning and cross-strapping of subsystem elements (sensors, processing, and actuators) can help avoid 'string' wipe-out with single element failures. The interfaces with the computer can be dual or even triplex to create input monitoring and fault screening. Thus, fault isolation or containment at the lowest functional level is performed, with the absence of major reconfiguration.

4.3.5.2.3 Distributed Intelligence. Addition of distributed intelligence, i.e., logically smart sensors and actuators in addition to executive processors, is appropriate. This supports the overall concepts given above by providing two-way, all digital communications between sensors data and actuator feedback at the device point, and protection for communications paths between smart elements by lower message rates, and is an attribute of this approach. The means to perform a self-test in a transparent manner is greatly enhanced by distributed microprocessors which can assess device health at the local level and report to the EXEC. An ACS is by its nature a complete system with several hierarchical levels and the need for feedback to function properly. Thus smart, two-way connections are natural for an increasingly sophisticated set of autonomous requirements.

4.3.5.3 DSCS III/Voyager Attitude Control Software Comparison. An initial assessment of the existing DSCS III attitude control software has yielded several points of comparison and contrast with the Voyager Attitude and Articulation Control Subsystem (AACS) on-board software. Section 4.3.5.3.1 discusses these comparisons. Section 4.3.5.3.2 provides an example of a possible on-board fault protection routine for DSCS III.

4.3.5.3.1 DSCS III/Voyager Attitude Control Routine Comparisons. Table 4-6 lists several DSCS III functions/routines with their analog in Voyager. The Voyager routines mentioned are a portion of the fault protection software onboard in both the AACS and CCS. The word count listed represents an approximate count of words for these fault protection routines, and may be used to represent a rough estimate of resources needed to implement these functions on DSCS III. The Voyager routines are described in more detail in Appendix B.

Figure 4-14 is a top/level flow diagram of the Voyager AACS software with special emphasis on the flow of the fault protection software. Figure 4-15 is a similar flow diagram for the DSCS III AACS software. Included in that figure are top level flows for the module DCNTRL and a portion of its update processing. These figures are included as a reference for the routines mentioned in Table 4-6.

4.3.5.3.2 Example of a DSCS III Attitude Control/Propulsion Function Fault Protection Routine: Reaction to Thruster/Plumbing Anomaly.

(1) DSCS:

After the normal orbit mode is obtained, the DSCS III spacecraft fires thrusters in two circumstances: (1) to perform a stationkeeping maneuver and (2) to periodically unload reaction wheels (perhaps twice per orbit). The thruster control laws process momentum estimates and commanded or nominal stored increment rates about each axis into thruster pulse counts. In addition, certain deadband values are computed and used to scale these pulse counts. In any event the pulse counts are used in commanding the thrusters to fire. In case of stationkeeping, some additional processing of the pulse count is done based on the mode of spacecraft operation. Also the stationkeeping timer is accessed and updated.

Table 4-6. DSCS III ACS and Voyager AACs Analogous Routines

PRESENT DSCS-III	SERVICE	VOYAGER	SERVICE	WORDS
TRPSER	Trap Error Service: Illegal instruction, Odd Address, Time-out from I/O port, Software from overflows	Error Interrupt (AACs) Error Routine (CCS)	Reset and Reinitialize critical variables Reset, Reinitialize and save special error data, also initiate power reset	80 230
Command Decoder CMDSER	Decode commands and perform sequence checking	CCS Command Interpreter Routine (AACs)	Decode commands from CCS and perform sequence and circumstance checks	200
Sun or Earth loss DCNTRL Blocks 4, 5	Await ground commands	Celestial Sensor Logic Routine (AACs)	Monitors sensors and enables/ initiates recovery sequences	200
Thruster firing DCNTRL Block 12	Compute thruster pulses from momentum about axes. Initiate commanded thruster pulse for ΔV . Ground monitors performance.	TCAPU Routine (AACs)	Monitor thruster pulses against limits and check angles about each axis. Anomalies serviced by plumbing swaps and/or spacecraft safing	95

Table 4-6. DSCS III ACS and Voyager AACS Analogous Routines (Cont'd)

DSCS-III	SERVICE	VOYAGER	SERVICE	WORDS
CPU Performance	Ground Monitors through telemetry	FCP Hardware Self test (AACS) (CCS)	Processor health check initiates swap to redundant units	20
		Memory refresh (AACS)	Monitor processor perfor- mance, initiates swap to redundant units,	30
		Power Code Processor (AACS)	Monitor transmission link between AACS and CCS, initiates swap to redundant units	65
Power Supply to CPU	Ground monitors through telemetry	Power supply fail routine (AACS)	Monitors power supplied to CPU and initiates swap to redundant units	20
		PWRCHK (CCS)	Monitors power interface and reconfigures to safe load	167
GROUND Analysis of faults	Ground views through standard telemetry (though at two different rates)	Omen power codes (AACS) (CCS)	Transmitted to CCS from AACS causes saving of next few power codes	40 (AACS) 40 (CCS)

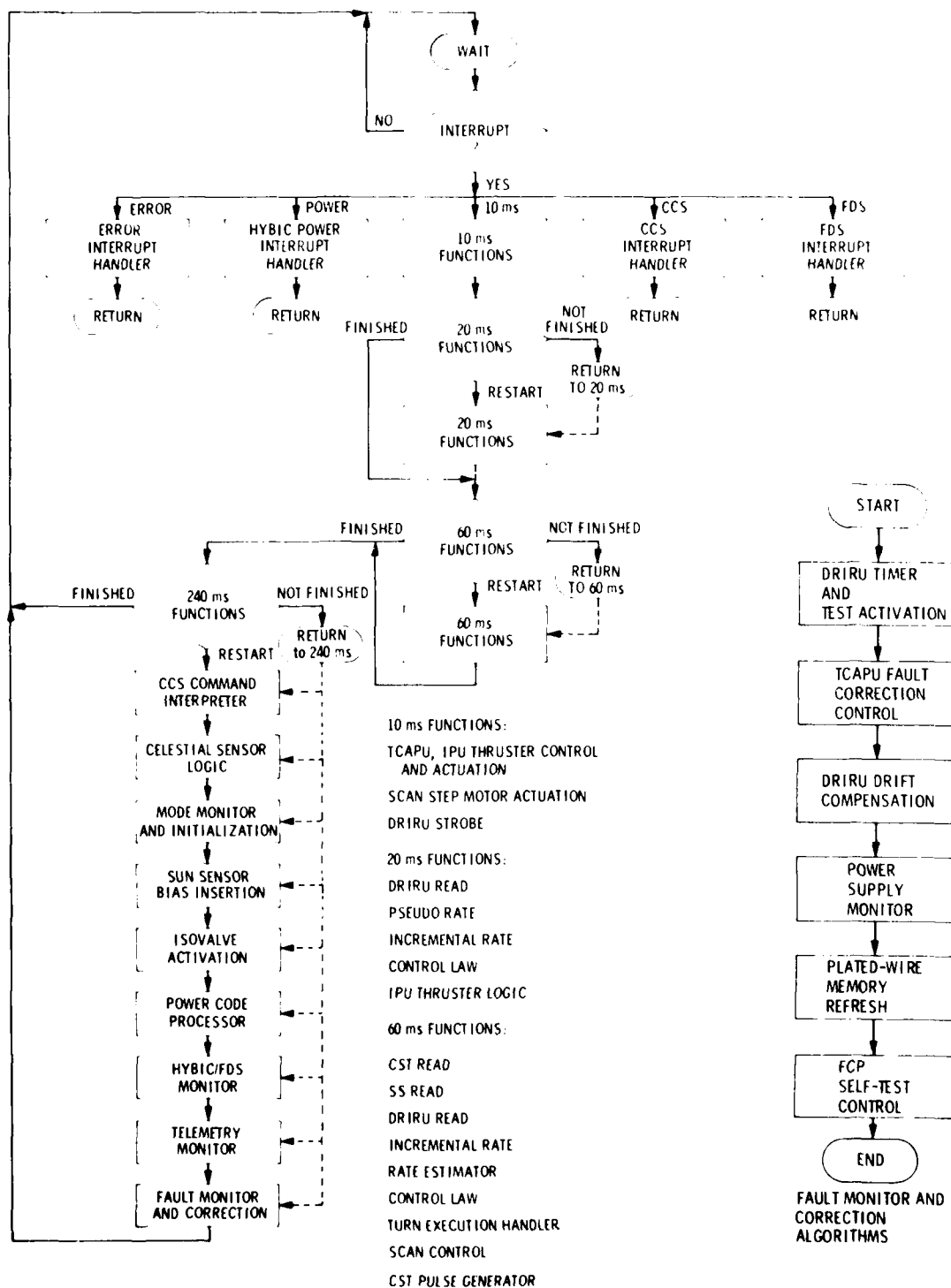


Figure 4-14. Voyager AACS Software Fault Monitor and Correction Algorithms

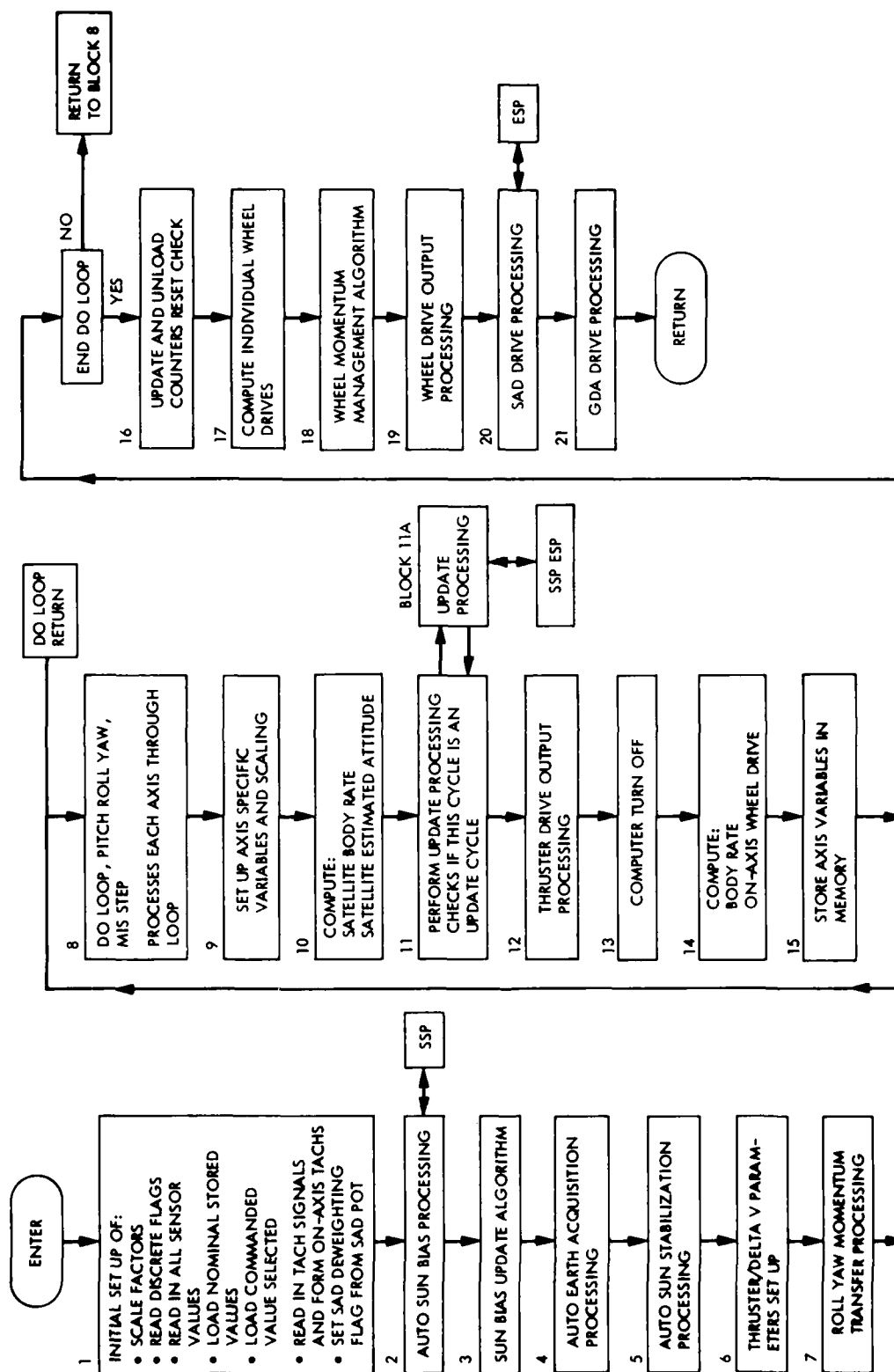


Figure 4-15(b). DCS III ACS Software: Attitude Controller (DCNTRL) Software Module Overall Block Diagram

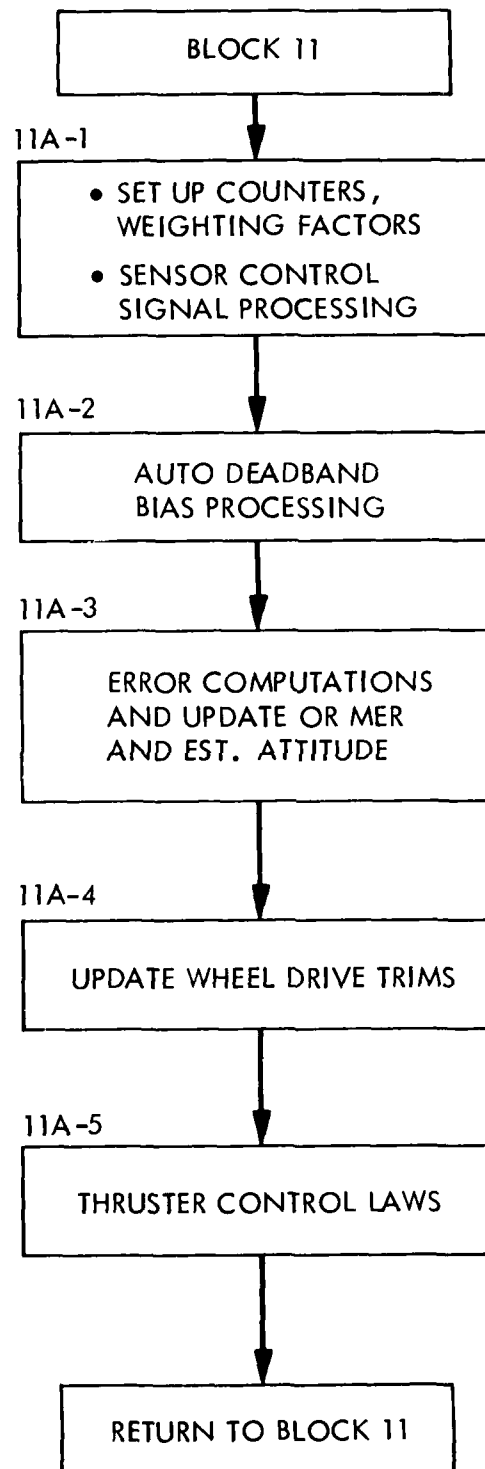


Figure 4-15(c). NSCS III ACS Software DCNTRL:Block 11A Top Level Flow of Update Process

However, the performance of the thruster firing and the resulting attitude of the spacecraft is never monitored by the on board software. The results are only visible in telemetry transmitted to the ground. Thruster performance anomalies (e.g., plume impingements) or plumbing anomalies (e.g., thruster stuck open/closed failures) must be deduced from this telemetry, and corrected by ground intervention. Such telemetry would likely be monitored by the ground during stationkeeping maneuvers. During auto wheel unloads, the autonomous operation does not rely on ground monitoring and control of thruster performance.

(2) Voyager:

After computation of the thruster pulses by the control law routine, the Voyager software routine TCAPU checks both the accumulated pulses against limits and the spacecraft angular displacement from each axis. If these checks fail, the first action (depending on mode) will be to arrange for a swap of thrusters. The Voyager configuration of thrusters and latching valves allows thruster pair isolation (as opposed to the configuration of DSCS III). Failed thruster pairs may be replaced by redundant pairs (in sets of P/Y and R) without swapping an entire branch of thrusters (see Figures 4-16 and 4-17).

In any case angle and pulse limits are widened, allowing the AACS sufficient margin to recover. If the checks of pulse and angle limits again fail, the circuitry of the AACS is swapped. This clears all memory of previous thruster swaps.

In case of a TCM (trajectory correction maneuver) the maneuver is aborted and a return to a safe mode of spacecraft operation is effected.

The Voyager spacecraft experienced several problems which lead to the exercising of the TCAPU fault protection logic. In some cases the tripping of this logic and the exercising of recovery sequences was unnecessary as judged by ground analysis of the problem. But whenever the TCAPU logic was used the spacecraft was autonomously returned to a safe mode of operation.

This kind of fault recovery and spacecraft safing is not present on DSCS III. The sensing and recovery from thruster faults must be accomplished by ground analysis of the fault, and timely transmission by the ground of commands to the spacecraft.

(3) Autonomous DSCS III Algorithm for Thruster Fault Protection During R/W Unloading.

In an attempt to incorporate fault protection for thruster firing during reaction wheel unloading on DSCS, the

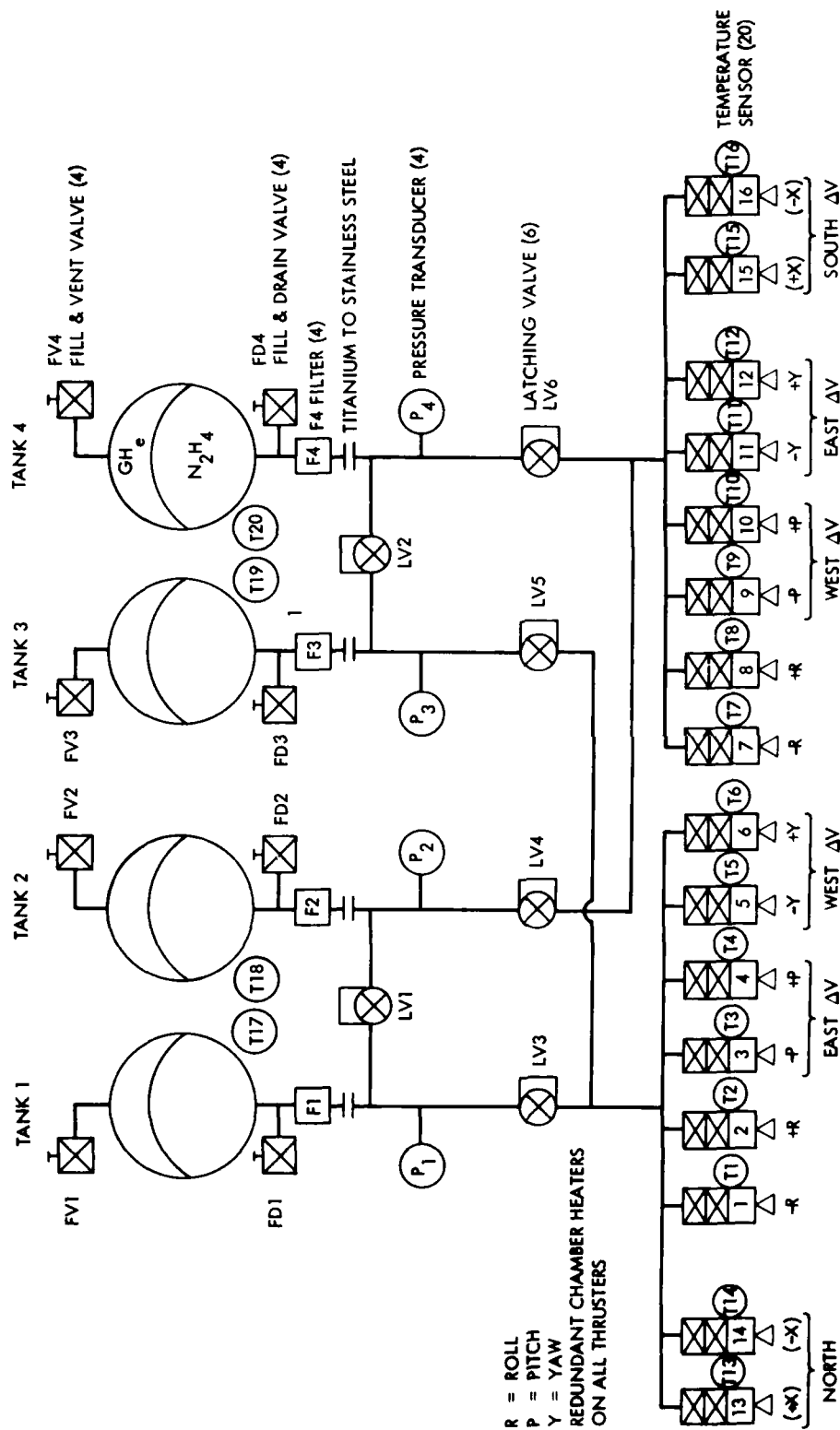


Figure 4-16. DSCS III Thruster Configuration

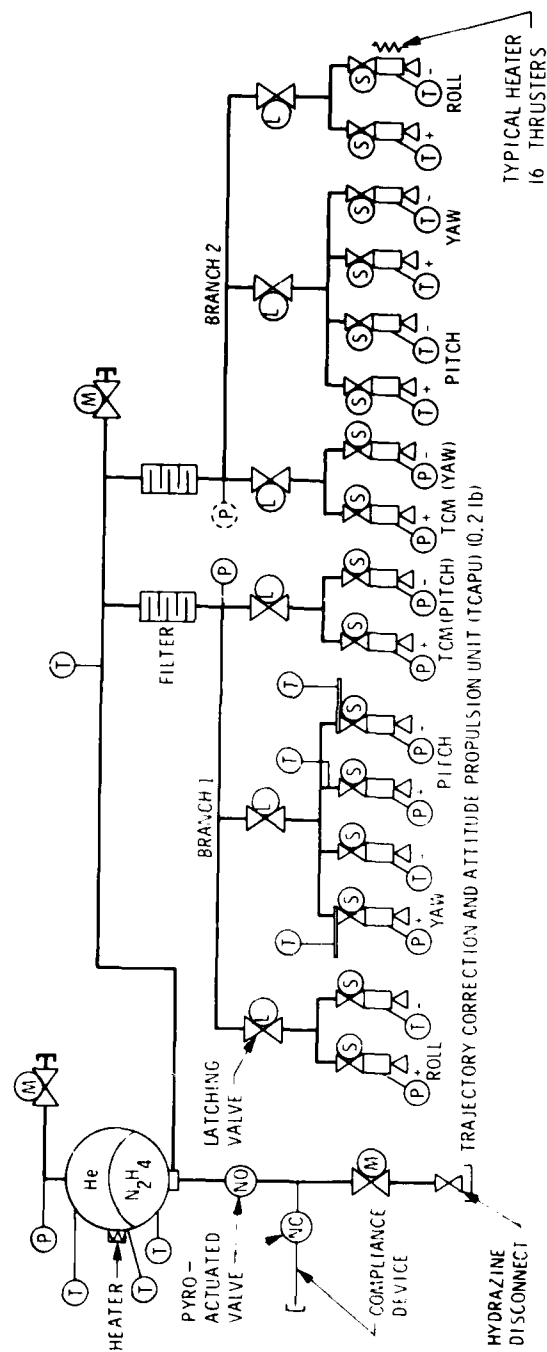


Figure 4-17. Voyager Propulsion Subsystem

following algorithm is offered. The assumptions for the algorithm are that the spacecraft is in the normal orbit mode, that the nominal update time of 16 sec is being used, and the software has access to I/O ports which allow configuring to redundant units.

Figure 4-18 is a flow of this algorithm. As a point of contrast Figure 4-19 is provided which is a data flow of a portion of the TCAPU routine associated with thruster firing fault protection in modes other than turns or TCM.

The algorithm in Figure 4-18 is initiated during the update cycle with an enabling flag which is set either by tachometer readings on an axis of above 80% of saturation, or as a result of ground command. In addition to setting the enabling flag, a timer for the sequence of thruster firings is set and the wheel unloading sequence is initiated. On subsequent entries to this routine the pulse count computed from the tachometer readings and the estimated momentum about the axis is compared against limits established for this update period or perhaps limits computed as an average over a previous wheel unload. In any event, a pulse count over the limits initiates a swap of thrusters.

The timer set to perform the unload directs control to initiating the firing sequence. Once the timer is up a performance check of the sequence is initiated based on the present tachometer values. An attempt to remedy a problem at this point involves a swap of thrusters (if this has not already been done) and a reinitiation of the unload sequence. A swap of circuitry and entry to another fault isolation algorithm will be necessary if a thrusters swap has not helped.

The tachometer limits check incorporates a portion of the ground procedure for an ACS health check (see procedure 6.1-2 DSCS Orbit Operations handbook).

In addition, the tachometer limits check should catch thruster-open anomalies and eventually lead to a thruster swap. The pulse count limits check should catch thruster closed anomalies and also lead to a thruster swap.

This routine would comprise approximately 75 words and be executed roughly within 1 ms and 1.5 ms with the current CPU on DSCS III. Since this routine must be executed for each axis, the total execution time may be 3 ms to 4.5 ms. However, this routine need only be executed during an update cycle (once each 16 sec) and only during thruster firing for a reaction wheel unload.

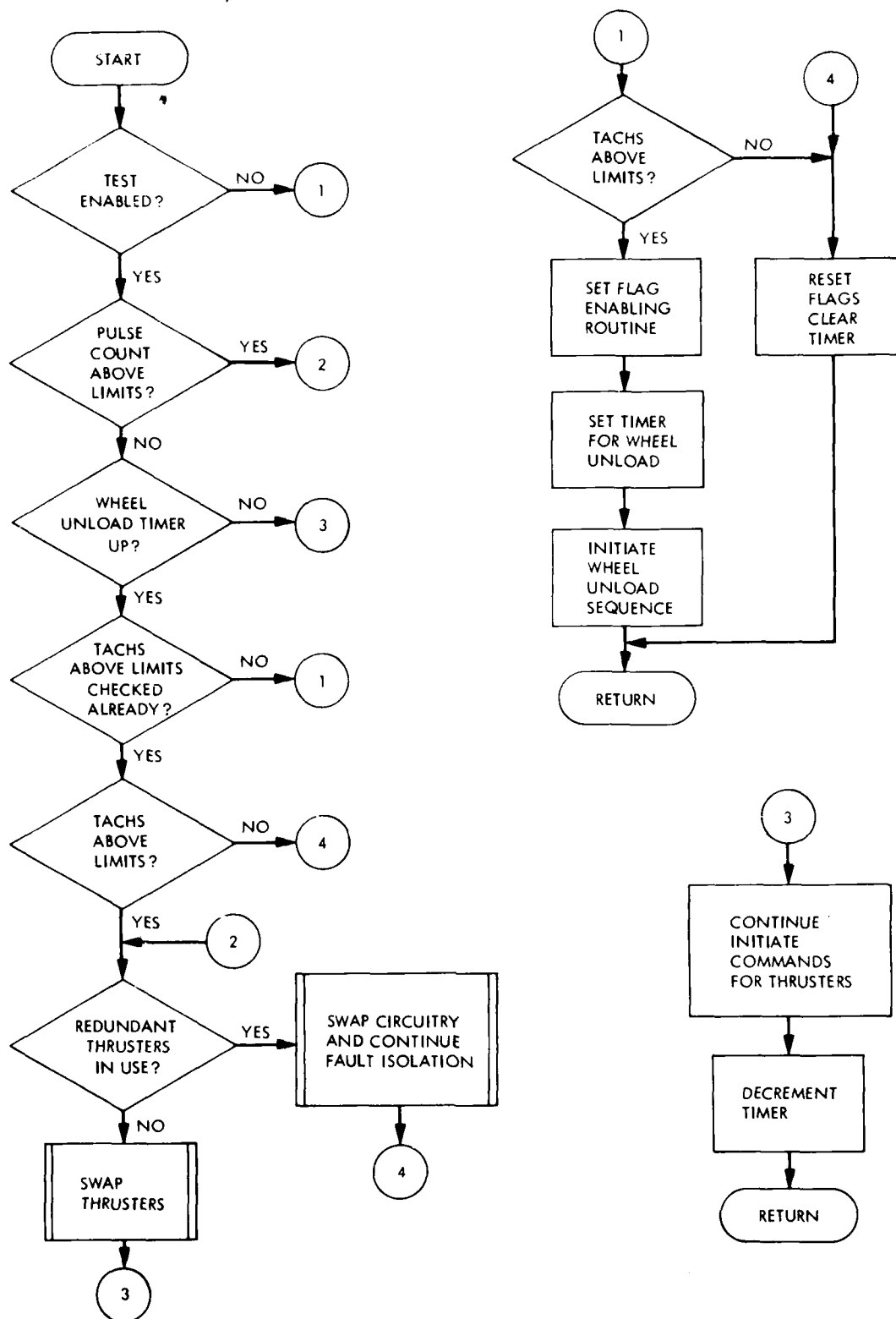


Figure 4-18. Example DSCS III Thruster Fault Algorithm for Autonomous Reaction Wheel Uploading

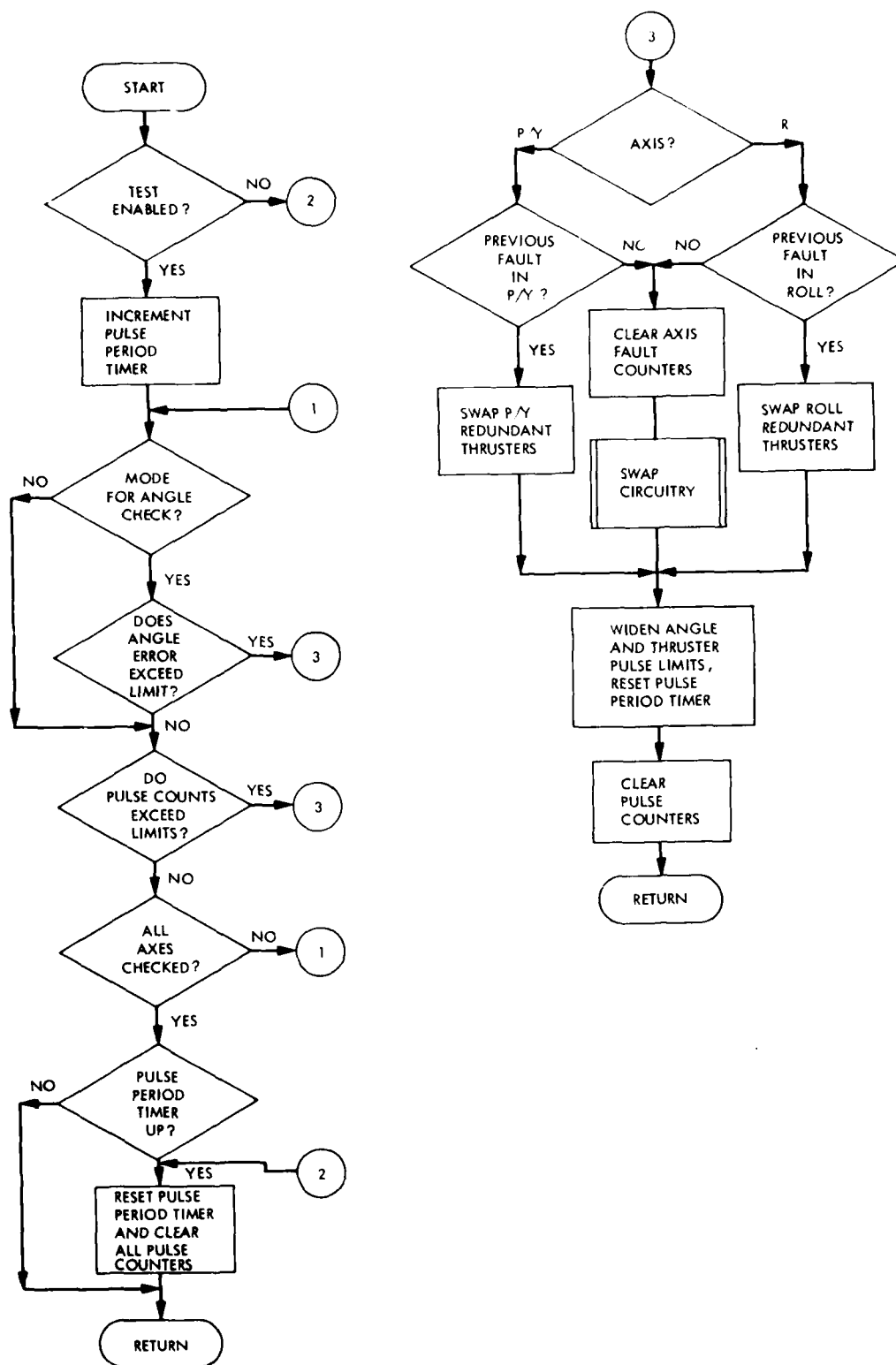


Figure 4-19. Voyager TCAPU Routine

4.4 AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF THE S/C THERMAL CONTROL FUNCTION - LEVEL 2/CATEGORY I*

Figure 4-20 is a Thermal Control Maintenance functional hierarchy.

In order for DSCS III to be made autonomous for 60 days without degradation, and six months with some degradation, the thermal control subsystem must:

- (1) Develop a self-check capability on heater circuits and other active thermal control subsystem components.
- (2) Determine logic requirements on the thermal control subsystem to detect failures and to implement corrective action. This includes developing autonomous operation of the battery recharge system.
- (3) Determine requirements for the autonomous operation of the navigation and attitude control subsystem. This involves all requirements for preconditioning of the propulsion subsystem and thermally active (e.g., reaction wheel) components of the attitude control subsystem that involve the vehicle thermal control subsystem.

Only two modifications to the DSCS III thermal control system are necessary to provide autonomous integrity maintenance. These can be implemented with a minimum of impact on the existing thermal control subsystem.

A system to detect a failure of the heater circuit and take appropriate action must be implemented. The system must also be able to detect a failure of the overtemperature thermostat and take corrective action.

The current design has no provisions to de-activate components of the survival system in the event of a failure of a thermostat. Thus, a system similar to the control system fault detection system must be implemented.

*By R. N. Miyake and J. A. Plamondon

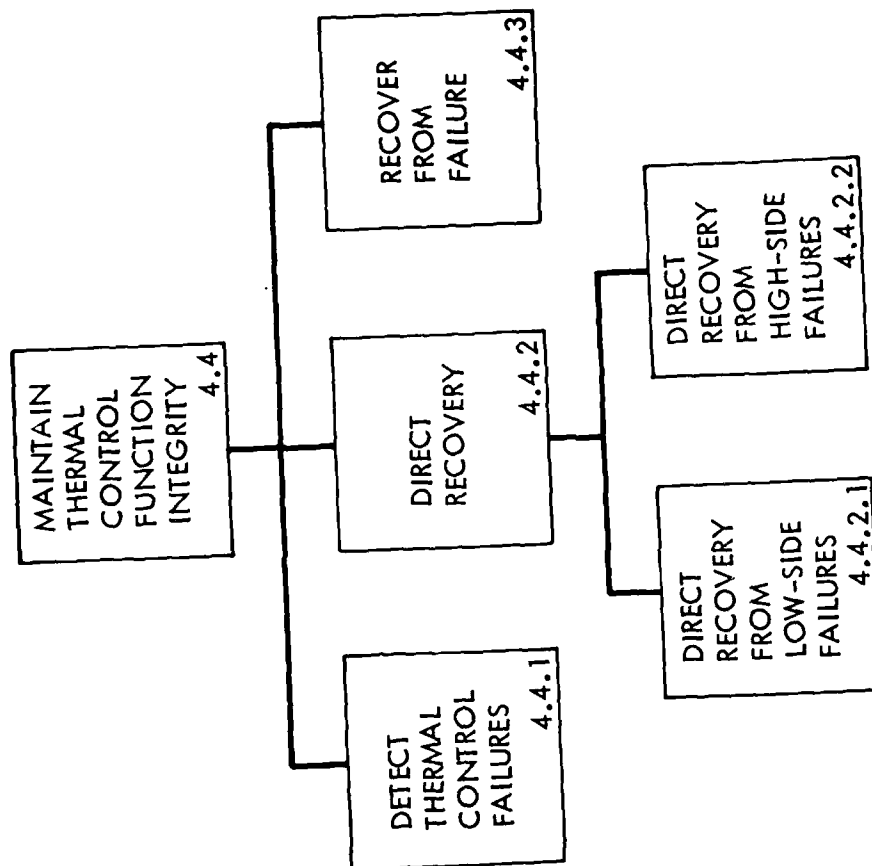


Figure 4-20. Thermal Control Integrity Maintenance Functional Hierarchy

4.5 AUTONOMOUS OPTIONS FOR MAINTAINING INTEGRITY OF S/C CONTROL AND MONITORING FUNCTION

To achieve autonomous integrity maintenance of the S/C control and monitoring function the fault detection and fault correction functions described in Section 4.5 of Volume II must be moved from the ground to the spacecraft.

As described in Section 4.5 of Volume II, the fault detection function involves the acquisition and analysis of spacecraft health information. The same telemetry stream accessed at the Remote Tracking Station (RTS) on the ground, to extract spacecraft health information, is available on the spacecraft. In fact, the requirements on the RTS to demodulate and decrypt the telemetry signal are not required on the spacecraft if the telemetry data stream is accessed at the output of the TT&C MTU prior to encryption. Interfacing of a microprocessor and associated memory capability to the output of the MTU would allow processing and status evaluation of selected measurements relating to the TT&C MTU, RTU, and CD performance. Limit checking and identification of apriori defined failure modes could be performed leading to the selection of stored software routines for corrective action. The stored software routines would be executed under microprocessor control resulting in the generation of plain text commands identical in format to those that would have been generated on the ground for the same failure mode assessment. The plain text commands would then be routed directly to the input of the TT&C CD as if they were coming from a decryptor. From there, the CD would perform its designated functions of processing and issuing the commands for redundancy switching execution.

Autonomous maintenance of the telemetry transmission from S/C to ground, and of the S-Band command link, are necessary to meet the goals of always having a fault protection audit trail available to the ground, and always having commandability of the spacecraft by the ground. These functions could, alternatively, be performed by the SHF telemetry and command link, which normally is used to control and monitor the payload functions. This solution would place an additional load on the payload control ground system, as described in Section 5.2.3.

Figure 4-21 shows the Maintain S/C Control and Monitoring functional hierarchy. Sections 4.5.1 and 4.5.2 describe, functionally, autonomous S/C control and monitoring maintenance. Section 4.5.3 presents options for implementing autonomous maintenance of the on-board portions of the S/C control and monitoring functions.

4.5.1 Maintain Telemetry Function

In order to autonomously maintain the integrity of the telemetry function, the maintenance of three functions must be carried out on board the spacecraft: 1) information acquisition maintenance, 2) telemetry generation maintenance, and 3) telemetry transmission maintenance. The remaining

By W. E. Arens and S. O. Burks

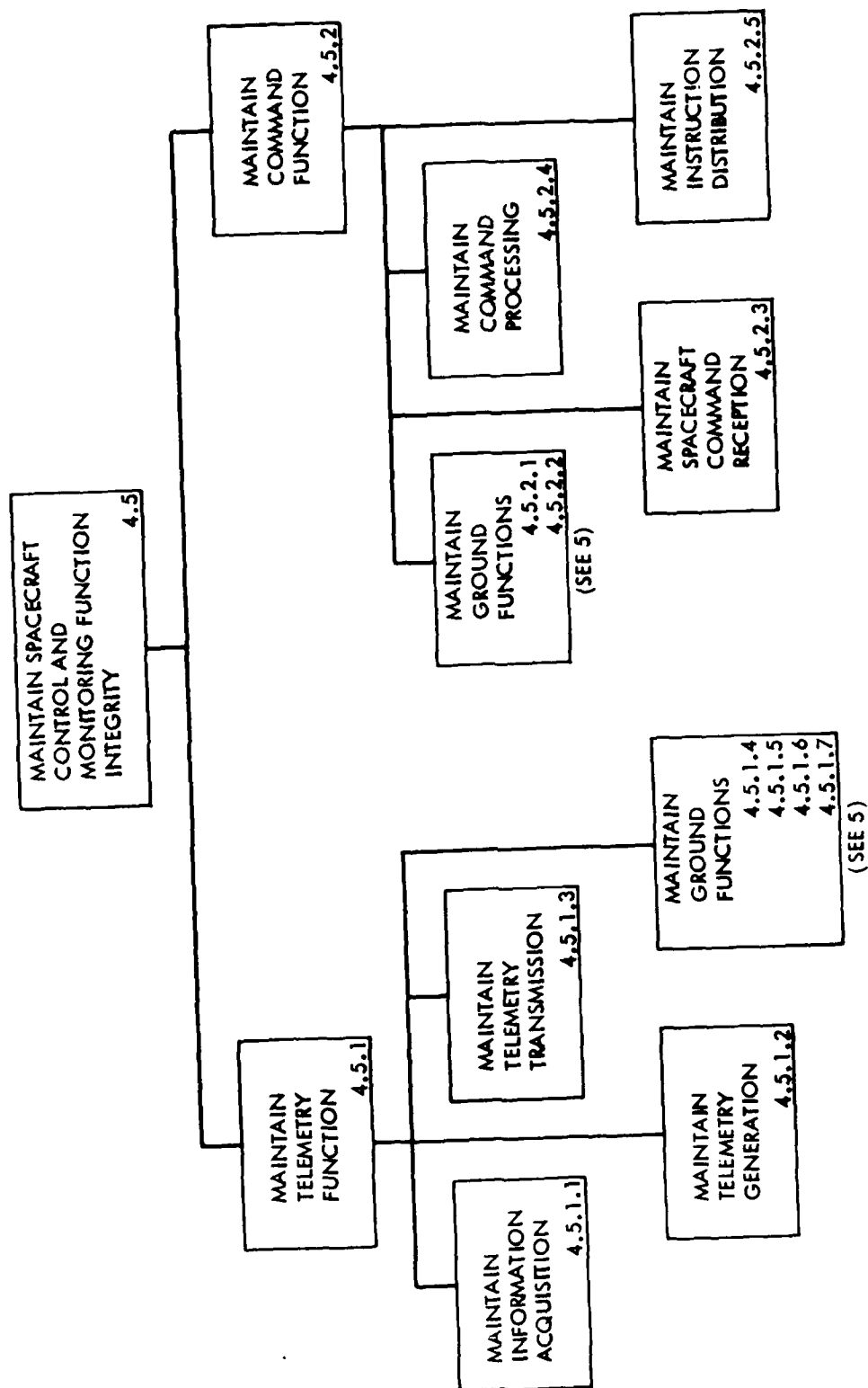


Figure 4-21. S/C Control and Monitoring Integrity Maintenance Functional Hierarchy

functions are inherently ground functions and will remain so although the nature of these functions will change, as described in Section 5. They are: 1) receive telemetry, 2) process telemetry, 3) distribute telemetry, and 4) analyze telemetry. The autonomous spacecraft will have to assume the functions of data reception (from internal sources), processing, distribution and analysis as required to carry out the spacecraft functions without ground intervention. However, if ground executive control is to be maintained the ground will have to receive information from the spacecraft and process it to determine if control of the autonomous features should be exercised; therefore, for an autonomously maintained spacecraft the process of handling telemetry on the ground will be to provide monitoring of the autonomous spacecraft features. Only the on-board functions of telemetry maintenance will be discussed in this section. Therefore, Sections 4.5.1.4 through 4.5.1.7 are not discussed here.

4.5.1.1 Maintain Information Acquisition - Level 2/Category I. Autonomous maintenance of this function will require on board analysis and switching of redundant elements of the MTU and RTU. This function must be maintained in order for on-board integrity maintenance of the spacecraft as a whole to be performed (see Section 4.1).

4.5.1.2 Maintain Telemetry Generation - Level 2/Category I. Autonomous maintenance of this function will require on board analysis and switching of redundant elements of the MTU and RTU. This function must be maintained in order for on board integrity maintenance of the spacecraft as a whole to be performed (see Section 4.1).

4.5.1.3 Maintain Telemetry Transmission - Level 2/Category II. All of the switching activities described in Section 2.4.1.3 are for providing the telemetry service. There are no "automatic" features on board to correct a telemetry failure.

The addition of telemetry transmission autonomy to the DSCS III spacecraft requires some assumptions about the utilization of the telemetry functions for operations. If it is assumed the telemetry function is only used when requested by the ground, "maintenance" autonomy is not really needed for X or S-Band telemetry on board the spacecraft. Whenever a command link is used to activate a spacecraft function that command link can be used to activate telemetry. That command link could be used to activate failure routines if there had been telemetry failure. Even the most complex autonomous spacecraft telemetry maintenance features can be done better on the ground with command link availability. However, if it is assumed that there is a need for periodic or continuous telemetry without ground intervention, then some form of autonomy is required to correct failures.

4.5.2 Maintain Command Function

In order to autonomously maintain the integrity of the command function the maintenance of three functions must be carried out on the spacecraft: 1) maintain S/C command reception, 2) maintain command processing, and 3) maintain instruction distribution. The remaining functions are inherently ground functions and will remain so, although the nature of these

ground functions will change, as described in Section 5. They are: 1) generate commands and 2) transmit commands. For an autonomously controlled spacecraft, the ground command function will be for executive control of the autonomous features. Only the on-board functions of command maintenance will be discussed in this section, and therefore Sections 4.5.2.1 and 4.5.2.2 are not discussed.

4.5.2.3 Maintain S/C Command Reception - Level 2/Category II. The current DSCS III S/C design includes some pseudo-autonomous features for X and S-Band command reception. It is pseudo-autonomous because the ground station has to change transmission frequencies to use the redundant command channel, if there is a receiver/detector failure. Both S and X band command paths use frequency diversity to select the receiver path. Decoders are selected by a command preamble word. Decryptors are selected by having a ground command link.

There are several possible approaches to providing more autonomy to the DSCS III command link. They range from relatively simple to relatively complex. They are discussed in Section 4.5.3. The key assumption for autonomy of the command function is that command capability be available to use within a "reasonable" length of time. This means that the X and/or the S-Band command link should be operational when ground based operations need to command the spacecraft.

4.5.2.4 Maintain Command Processing - Level 2/Category I. Autonomous maintenance of this function will require on-board analysis and switching of redundant command decoder blocks. This function must be maintained in order for on-board integrity maintenance of the spacecraft as a whole to be performed (see Section 4.1).

4.5.2.5 Maintain Instruction (Command) Distribution - Level 2/Category I. Autonomous maintenance of this function will require on board analysis and switching of redundant command decoder blocks. This function must be maintained in order for on-board integrity maintenance of the spacecraft as a whole to be performed (see Section 4.1).

4.5.3 Options for Implementation of Autonomous Maintenance of the S/C Control and Monitoring Functions

An approach to redundancy management for integrity maintenance using a Redundancy Management Subsystem (RMS) as described below, appears potentially feasible for achieving a significant level of autonomy for the TT&C MTU, RTU, and CD functions. Assuming the use of existing flight qualified components and establish techniques, an implementation characterized by minimal weight, size, power, cost, and risk should be possible. Furthermore, a preliminary assessment of the approach indicates virtual transparency to the current DSCS III spacecraft design.

In addition, options are presented for autonomous integrity maintenance of the S/C command reception and telemetry transmission functions.

These functions are required for ground executive control and for audit trail access by the ground, but are not required for 60 day/6 month autonomy without ground intervention.

4.5.3.1 A Redundancy Management Subsystem Option for the TT&C Subsystem. This section describes a Redundancy Management Subsystem (RMS) which can handle the fault detection/fault correction functions of the TT&C subsystem. This section emphasizes redundancy management for the information acquisition and processing, and command processing and distribution functions which are performed by the TT&C MTU, RTU and CD. The functional characteristics of the RMS described in this section include making the MTU, RTU and CD fault tolerant to single point failures.

A functional hierarchy for the DSCS RMS is provided in Figure 4-22. This figure describes four functional areas. They are 1) health information acquisition, 2) health information analysis, 3) command generation, and 4) command distribution. An annotated flow diagram for these functional areas is given in Figure 4-23. Narrative descriptions for each block of the functional hierarchy are provided in the following paragraphs. In support of the narrative descriptions of the RMS functions, Figures 4-24, and Tables 4-7 and 4-8 are provided. Figure 4-24 defines the functionally redundant block levels for the TT&C MTU, RTU, and CD.

Table 4-7 defines 1) some possible fault indications, 2) associated diagnostic software subroutine evaluation results based upon these indications, 3) fault isolation conclusions from these results, and 4) corresponding fault correction actions. Table 4-8 defines required functional elements of the RMS in order that it accomplish its designated functions.

It should be emphasized that this discussion only treats the fault detection and correction functions required for the TT&C MTU, RTU, and CD. It also implicitly assumes that the RMS is inherently fault tolerant with respect to its own internal single-point failures.

4.5.3.1.1. Acquire Health Information. TT&C MTU, RTU, and CD health information is acquired by the RMS from both 1) the output telemetry data stream from the MTU, and 2) special-purpose diagnostic signal responses. To eliminate the need for adding sensors to the current DSCS III TT&C subsystem design, simulated sensor signals are generated by the RMS and provided to designated redundant functional blocks for diagnostic purposes. The resultant signal responses may be either returned 1) via allocated measurement locations in the output telemetry stream, or 2) directly to the RMS under the category of special purpose.

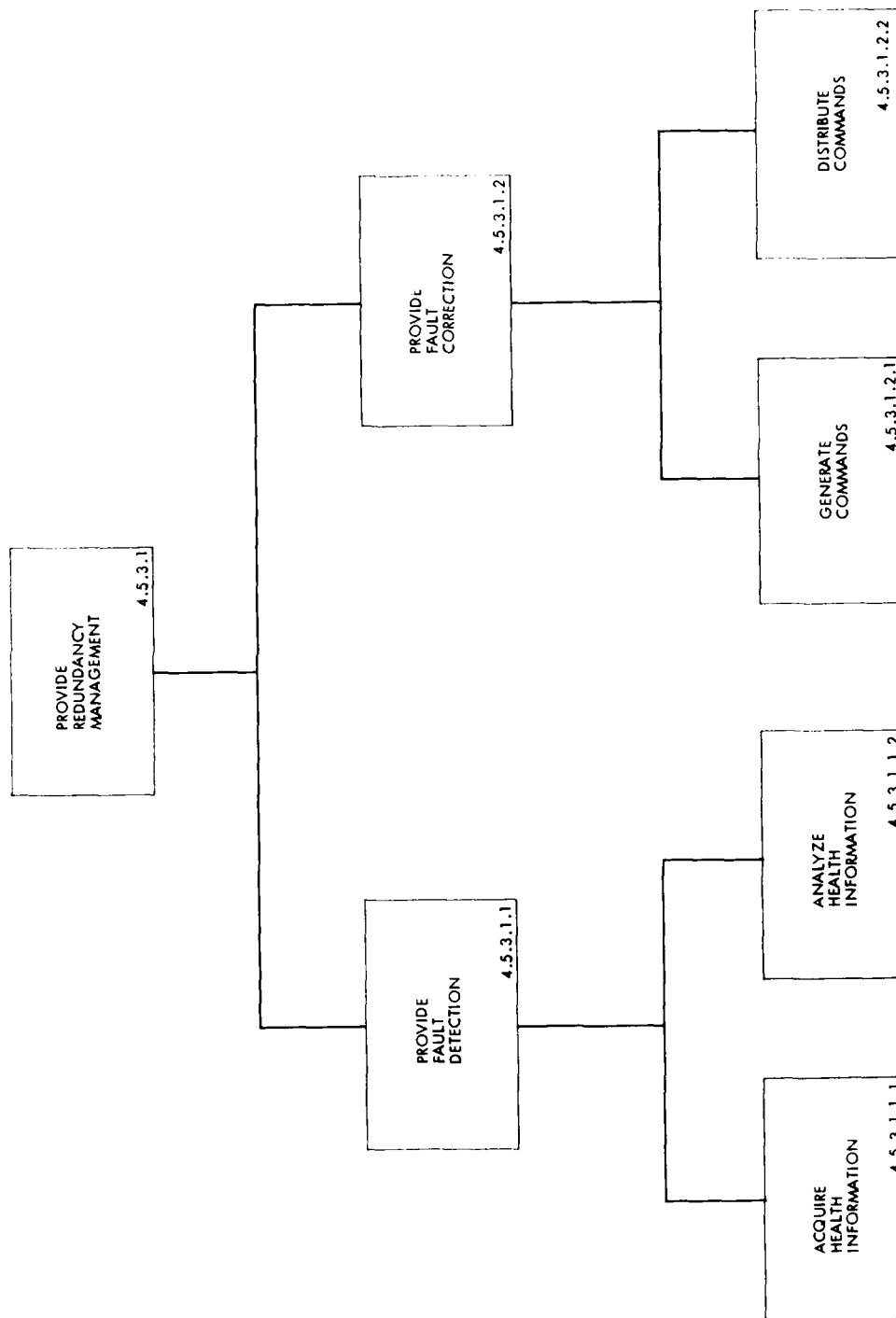


Figure 4-22. DSCS RMS Functional Hierarchy

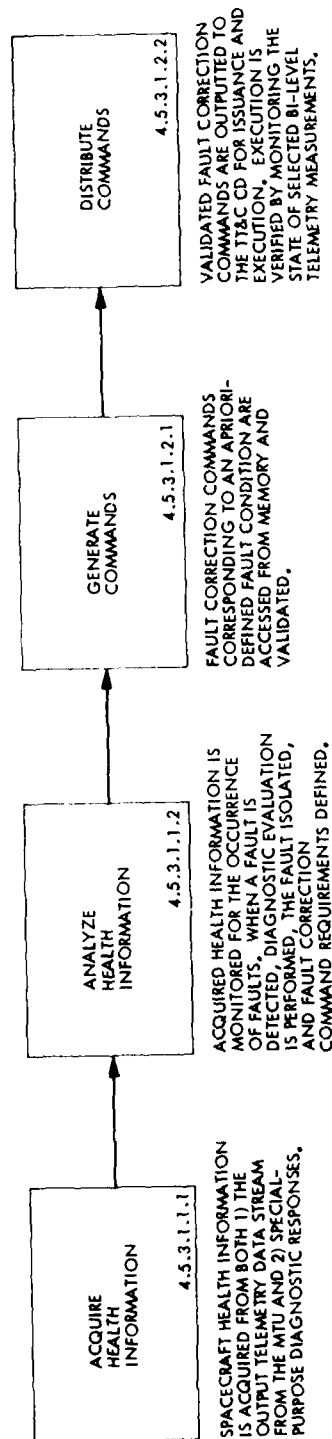


Figure 4-23. DSCS RMS Functional Flow and Description

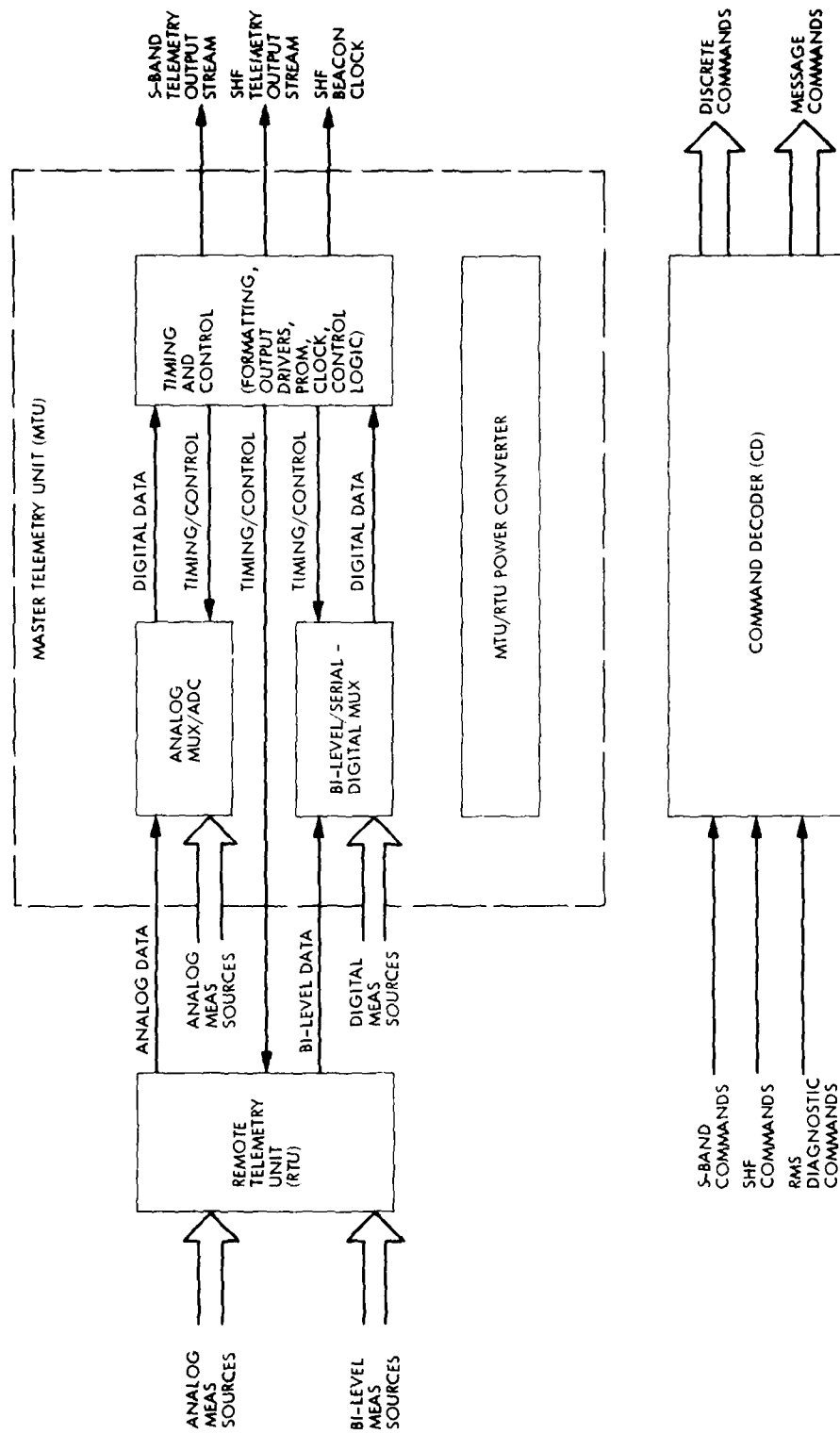


Figure 4-24. DSCS TT&C MTU, RTU, and CD Functionally Redundant Blocks

Table 4-7. DSCS RMS Fault Detection/Fault Correction Matrix for TT&C MTU, RTU, and CD

DETECTED FAULT	SUBROUTINE ANALYSIS RESULTS	FAULT ISOLATION	FAULT CORRECTION
$\begin{pmatrix} E_1 = \text{RMS ANALOG REFERENCE VOLTAGE MEAS} \\ E_2 = \text{RMS DIGITAL REFERENCE VOLTAGE MEAS} \end{pmatrix}$	E_1 FROM RTU ANALOG MUX _____ BAD E_1 FROM MTU ANALOG MUX/ADC _____ OK E_2 FROM MTU B-L/S-D DIG MUX _____ OK MTU VOLTAGE LEVELS _____ OK	RTU	SWITCH TO REDUNDANT RTU BLOCK
	E_1 FROM RTU ANALOG MUX _____ OK E_1 FROM MTU ANALOG MUX/ADC _____ BAD E_2 FROM MTU B-L/S-D DIG MUX _____ OK MTU VOLTAGE LEVELS _____ OK	MTU ANALOG MUX	SWITCH TO REDUNDANT MTU ANALOG MUX/ADC BLOCK
	E_1 FROM RTU ANALOG MUX _____ BAD E_1 FROM MTU ANALOG MUX/ADC _____ BAD E_2 FROM MTU B-L/S-D DIG MUX _____ OK MTU VOLTAGE LEVELS _____ OK	MTU ADC	
	E_1 FROM RTU ANALOG MUX _____ OK E_1 FROM MTU ANALOG MUX/ADC _____ OK E_2 FROM MTU B-L/S-D DIG MUX _____ BAD MTU VOLTAGE LEVELS _____ OK	MTU B-L/S-D MUX	SWITCH TO REDUNDANT MTU BI-LEVEL/SERIAL-DIGITAL MUX BLOCK
	E_1 FROM RTU ANALOG MUX _____ BAD E_1 FROM MTU ANALOG MUX/ADC _____ BAD E_2 FROM MTU B-L/S-D DIG MUX _____ BAD MTU VOLTAGE LEVELS _____ OK	MTU TIMING AND CONTROL	SWITCH TO REDUNDANT MTU TIMING AND CONTROL BLOCK
	E_1 FROM RTU ANALOG MUX _____ BAD E_1 FROM MTU ANALOG MUX/ADC _____ BAD E_2 FROM MTU B-L/S-D DIG MUX _____ BAD MTU VOLTAGE LEVELS _____ BAD	MTU POWER CONVERTER	SWITCH TO REDUNDANT MTU POWER CONVERTER BLOCK
	VERIFY CD WILL NOT ISSUE COMMAND	CD	SWITCH TO REDUNDANT CD BLOCK
	VERIFY INCORRECT PROCESSING IN CD	CD	
	FAILURE TO DETECT C_1 RESPONSE OR C_1 RESPONSE READING IN ERROR		
	C_1 = DIAGNOSTIC SELF-ADDRESSED COMMAND FROM RMS		

Table 4-8. DSCS RMS Functional Elements

<p>INPUT/OUTPUT (I/O)</p>	<p>PROVIDES INPUT AND OUTPUT INTERFACE WITH TT&C SUBSYSTEM</p>
<p>PROGRAMMABLE READ-ONLY MEMORY (PROM)</p>	<p>STORES EXECUTIVE SOFTWARE AND COMMAND ADDRESS TABLES</p>
<p>READ/WRITE MEMORY (RWM)</p>	<p>STORES FAULT DETECTION SOFTWARE ROUTINES; STORES FAULT CORRECTION COMMANDS; BUFFERS TELEMETRY DATA FOR STORAGE</p>
<p>NON-VOLATILE MEMORY (NVM)</p>	<p>STORES PERTINENT TELEMETRY DATA FOR SUBSEQUENT TRANSMISSION TO EARTH; STORES CRITICAL SOFTWARE ROUTINES FOR RELOADING RWM</p>
<p>CENTRAL PROCESSOR UNIT (CPU)</p>	<p>PROVIDES TIMING, CONTROL, AND DIGITAL PROCESSING FUNCTIONS</p>

The RMS generates and issues simulated signals to the MTU, RTU, and CD for diagnostic purposes.

A predefined analog reference voltage and its digital counterpart are provided by the RMS. Predefined plain-text commands are also generated by the RMS and provided to designated redundant functional blocks for diagnostic purposes. The resultant signal responses may be either returned 1) via allocated measurement locations in the output telemetry stream, or 2) directly to the RMS under the category of special purpose.

The analog reference voltage and its digital counterpart from the the RMS are connected via dedicated lines to appropriate input terminals of the TT&C MTU and RTU multiplexers. Dedicated lines are also provided for routing the RMS generated commands to the input of the TT&C command decoder (CD).

The analog reference signal connected to the MTU and RTU analog multiplexer inputs is continuously applied so that no further action is required prior to interrogation by the TT&C MTU and RTU, respectively. The serial-digital measurement signal, which is routed to the MTU bi-level/serial-digital multiplexer, is issued upon receipt of enable and clock signals by the RMS from the MTU. The diagnostic commands from the RMS are periodically issued to the TT&C CD at predetermined time intervals.

The RMS receives the output telemetry stream from the MTU and decommutates the measurements associated with the MTU, RTU, and CD performance. This includes the diagnostic reference signals routed to the MTU and RTU multiplexers from the RMS.

The RMS receives special purpose diagnostic information regarding the health of the TT&C CD, directly from the CD, over dedicated lines. This information is in the form of the presence or absence of valid commands issued to the RMS by the CD in response to the self-addressed commands routed to the CD from the RMS.

4.5.3.1.2 Analyze Health Information. Health information for the TT&C MTU, RTU, and CD is monitored using both measurements extracted from the output telemetry stream and special purpose diagnostic responses. This information is compared against acceptable measurement limit values and predefined performance criteria. When unacceptable measurement values and/or performance are detected, a fault occurrence is identified. Following detection of a fault occurrence, diagnostic evaluation is performed to identify the source of the fault. Once the fault has been isolated, the required corrective action is defined.

Digital processing circuits in the RMS are used to detect unacceptable signal information from both telemetry channel measurements and special-purposes returns associated with the TT&C MTU, RTU, and CD.

Verification of a fault occurrence in the TT&C MTU, RTU, or CD and location of the fault source to the redundant block level is accomplished by initiating and executing a selected software subroutine from memory based on the specific fault occurrence identified. A typical software subroutine might sequentially examine the status of selected apriori-defined measurements from the telemetry stream. The resultant combination of measurement status information would then enable a fault occurrence verification and a fault source identification to be achieved using RMS digital processing circuits.

The required correction for any apriori-defined MTU, RTU, or CD fault, which has been isolated to the redundant block level, is stored in software at a predefined MTU read/write memory address in the form of a redundancy switching command. RMS digital-processing circuits are used to access the appropriate correction command address from a table stored in PROM based on the designated fault source identification.

4.5.3.1.3 Provide Fault Correction. Commands for effecting the required correction for any apriori-defined fault occurring in a redundant block of the TT&C MTU, RTU, and CD are stored in software at addresses in an RMS read/write memory. By properly addressing the RMS read/write memory, a fully-formatted, plain-text command to be used for switching a designated functionally redundant block is accessed from the memory. The command is then routed to appropriate RMS digital-processing circuits for validation purposes prior to issuance.

A designated command for switching to a functionally redundant block to correct for an identified fault in the TT&C MTU, RTU, or CD is accessed from a read/write memory in the RMS under RMS microprocessor control.

Plain-text fault correction commands accessed from the RMS read/write memory are checked with respect to authenticity by appropriate RMS digital processing circuits. The command type and issuance address are interrogated and compared with the fault source identification information. Format and bit parity are also checked to insure command structure integrity. When validated, the command is routed to the TT&C CD. If rejected, appropriate RMS digital-processing circuits initiate a fixed contingency subroutine stored in PROM.

4.5.3.1.4 Distribute Commands. A redundancy switching command, which has been accessed and validated by the RMS in response to a verified fault occurrence, is outputted from the RMS to the input of the TT&C CD. The CD processes the command and issues it to its addressed destination where it is executed by replacing a faulty block with a functionally redundant equivalent.

When an issued fault correction command is executed by switching redundant blocks at its addressed destination, the resultant configuration change will be indicated by a bi-level measurement change-of-state in the output telemetry data stream. The RMS will detect this configuration change, thereby verifying execution of the command, by monitoring the state of the appropriate telemetry bi-level measurement.

4.5.3.2 Autonomous Options for Telemetry Transmission Function Maintenance. The on/off sequencer described in Section 2.4.1.3 just maintains a service on a periodic basis. It does not correct for failures unless other autonomous features are added. The items described below are failure detection techniques. All of these techniques below will require the telemetry function hardware to be "ON" periodically to test the hardware health. For this reason a sequencer will probably be required for any of the autonomous additions below.

4.5.3.2.1 RF Power Monitor. Since Mariner '64 the JPL Radio Frequency Subsystems (RFS's) have flown with a sensor diode on the output of the transmitter to detect the presence and level of RF power out. These sensors only detect the presence and level of RF power and not the quality of the telemetry data or the RF carrier. There are failure modes in the telemetry path that the RF detector would not detect (e.g., no telemetry modulation).

The implementation design at X or S-band is simple, i.e., a directional coupler, diode detector and conditioning circuits. Since the DSCS III already has RF output telemetry sensors, it probably is possible to use these sensors. The output of these sensors could be conditioned in such a manner that the TT&C subsystem could switch redundant elements or initiate a switch by another subsystem (e.g. the south power controller, the command decoder, or a "Redundancy Management Subsystem"). The technology for adding this function is readily available. However, testing this function at the spacecraft level is moderately difficult except for simple on/off RF tests.

4.5.3.2.2 Telemetry Failure Sensing. A system could sample the output of all TT&C telemetry using the RMS. The TT&C data could be sampled with pre-programmed actions occurring when the telemetry functions exceed established limits. Any system like this would need variable limits which could be set by ground command.

This system is relatively simple to implement from a hardware/software standpoint. However, there are numerous TT&C hardware failure modes that make even a simple on-board telemetry analysis difficult. Frequently it

is difficult for ground personnel to analyze the telemetry data and determine what happened. Implementing a good on-board failure sensing algorithm will be difficult with existing telemetry and probably even more difficult with additional telemetry (i.e., additional telemetry will increase the complexity of the failure modes which could be detected).

This system does have the advantage of not adding very much additional hardware to the TT&C subsystem. However, it does add a relatively complex subsystem: the RMS. There is a high level of engineering which goes with the RMS to determine the failure modes and telemetry limits. (For some unknown reason, RFS hardware does not fail in preconceived failure modes.).

4.5.3.2.3 Direct Telemetry Function Failure Sensing. The most straightforward method of detecting a telemetry function failure is to sense that function as it is examined and used on the ground. That is, receive and process the telemetry signal on the spacecraft. This would require the following types of functional elements.

- (1) RF probe sampling the transmitter output (possibly at the antennas). One each would be needed for S and X-band;
- (2) Strong signal receiver (one for S and one for X);
- (3) Data detector (one for S and X);
- (4) Decryptor (one for S and X);
- (5) Data decoder (common to S and X);
- (6) Data comparator sampler (common);
- (7) Data failure detector (common);
- (8) Failure management (common).

This direct sensing approach will detect failures in the telemetry functional hardware which could cause a loss or severe degradation of telemetry performance. The chief disadvantage of this approach is the addition of massive and power consuming hardware (6-12 Kgs and 10-20 watts). The mass and power increase would be governed by the chosen technology. The DSCS III TT&C technology possibly could weigh as much as 12 Kg and consume up to 20 watts. In some of the currently available technologies a telemetry failure detection system might weigh 6 Kg's and draw 10 watts or less.

There is probably no major technology development in this approach. However, if a very lightweight low power design is needed LSI, VLSI and RF LSI techniques would require use of relatively new technology.

This approach is easily tested whenever telemetry is on. It could be accomplished in parallel with other tests.

4.5.3.3 Autonomous Options for Maintaining Command Reception Function. There are several approaches for increasing the autonomy of the DSCS III S/C command reception function. In the event of a failure, these approaches could make the command channel available almost immediately upon request or available in seconds, minutes or hours. It will be assumed that the command channel has to be available with some "reasonable" time period for commanding. It will also be assumed that the preferable implementation would make available either the S-Band or the X-Band channel autonomously.

Based on current information, the X-Band command reception path should be used more than the S-Band. It is used for commanding the payload for user operations. Making the X-Band autonomous will probably require major surgery to the design. The single-point failures in the current X-Band command design will have to be eliminated. For example, the decryptor hardware might have to be "ON"; the switching between cross-strapped elements might have to change; and, the frequency standard reference to the TT&C subsystem from the Comm subsystem could require redesign. The amount of surgery depends on the autonomy approach chosen.

The following are possible additions to achieve more autonomy in DSCS III.

4.5.3.3.1. Use As Is Option. The current design can be commanded with a ground decision that one of the four possible command paths tried did not work and then selecting one or more of the other three paths (e.g., one S-band frequency path did not work, there are two X-Band and another S-Band path to try). This is a pseudo-autonomous operation because the ground personnel are involved in the selection of an alternate command channel by changing the ground transmission frequency. In a similar manner, the command decoder A or B can be selected by a command preamble word.

The existing design could be used but all of the redundant X-Band receiving elements could be left in "active" standby. This is basically item 4.5.3.3.2 below.

4.5.3.3.2 S-Band Common Frequencies. The S-Band command paths could be changed from different to identical frequencies. In this case, both command strings (A and B) could acquire the uplink command signal and output data to the decryptors. The decryptors could output up to two sets of command data to the command decoders (CD). The command decoder would now have to make a decision as to the validity of the command data. This validity decision could be accomplished by simply having a ground selected preamble word, as currently designed. However, for autonomous operation an on-board decision would have to be made to determine Command Decoder operability. Currently in the design, this decision is made by ground personnel and the selection is made by a CD preamble identification and a cross-strapping word in the command data. To have the decoder decide if it is working and to evaluate the quality of the command data is more complex than the current system. However, systems currently flying are implemented with all command elements in active standby and the decoder determining the validity of the command path. This approach would require redesign but the added logic and hardware should weigh less than 0.25 Kg and draw less than 0.25 watts. The technology is straightforward and the testing is simple.

4.5.3.3.3 X-Band Dual Carrier Transmission. The frequencies of the X-Band receive path cannot be made identical as in the case of the S-Band. The X-Band receive function shares active and passive RF components with the Comm subsystem. The Comm subsystem has to have different frequency channels. To acquire the X-Band channel quickly (as with S-band above) the ground stations could transmit "dual" carriers when a command is sent.

To achieve full autonomy in the event of a single failure, both of the Comm subsystem redundant 5 MHz frequency references would have to be on and their outputs supplied to the X-Band uplink hardware. Also, both decryptors would have to be in active standby. The input/output cross-strapping would have to change. The Command Decoder logic would be the same as (2) above in selecting the valid command channel. This approach will cause moderate but straight-forward redesign with about 0.5 Kg mass and 20 watt power increases. (The large power increase is a result of changing from passive to active standby.) The technology is straightforward and so is the testing for this approach.

4.5.3.3.4 Telemetry Failure Sensing. To sense a command function failure the TT&C MTU output telemetry data could be sampled (i.e., by the RMS). The TT&C telemetry data could be evaluated and appropriate actions taken when the data falls outside prescribed limits. Of course, the system would have to have command variable "alarm" limits.

This type of system is relatively simple to implement from a hardware and software standpoint. The technology and testing also are relatively simple. However, using existing DSCS III telemetry to accurately deduce a command function failure will be extremely difficult. The addition of more analog or digital monitor points to the design in order to determine failures more accurately would require a very large engineering effort to determine the failure modes and the telemetry limits for those failures. Furthermore, achievement of a satisfactory command failure algorithm is doubtful. There are too many failure modes in the TT&C analog and digital hardware which would make failure deduction difficult. Even with knowledgeable ground based operators reviewing telemetry data, failure detection and correction is not easy.

4.5.3.3.5 Direct Command Function Failure Sensing. The most straightforward method of detecting a command function failure is to test the command function as it is used. That is, use on-board hardware to send and process commands through the command functional elements. This approach would require the following type of functional elements:

- (1) RF probe for injecting a RF signal into the X and S-band receive paths (preferably at the antenna);
- (2) Command test transmitters at S and X-Band to inject a low level signal (3-10 dB above threshold) into the receive paths.

- (3) A command generator.
- (4) An encryptor.
- (5) Command validator at the command decoder output to determine the quality of the command channels.

This direct command failure sensing technique could detect failures in the command channel that might not easily be detectable by telemetry. The command test would have to be implemented on a cyclic basis with an inhibit when ground commanding is in process. It is about the only approach, other than sending a ground command, that reliably will detect a failure. It does have the disadvantage of adding a fair amount of hardware. It would probably add 5 to 10 Kg and 8 to 16 watts to the spacecraft weight and power requirements. The total mass and power increase would be governed by the technology chosen. Using the current DSCS design technology would require the higher values for mass and power. Some of the currently available technology could achieve the lower values. Unless exotic RF LSI and VLSI techniques are used the technology is straightforward. The testing of this approach is very simple.

4.5.3.3.6 Cyclic Command "Not to Switch". Several of the JPL S/C since the Viking Orbiter have used the "negative" command approach to switching command path elements. For this approach, if a command is not processed by the spacecraft within a selectable time period, a command failure algorithm will switch various elements until a valid command is received and processed. For a deep space mission where there generally is plenty of time available (hours to days), this is probably a satisfactory approach. For DSCS III this type of implementation may not be suitable. The command times and periods could be expected to be random. The link could be used more than once daily or possibly not for several months. The necessity of processing a command within a fixed period could result in switching perfectly good hardware if a command is not transmitted before the period was up. Since the ground stations could be out of service for months, the command algorithm could result in hardware switching continuously for extended periods. This switching could be detrimental to the hardware reliability. Another disadvantage of this approach is that in the event of a failure the command channel might not be available in a "reasonable" time period.

The chief advantage of this approach is that if a valid command is not processed, a relatively simple preprogrammed set of actions could be initiated. A rough estimate of TT&C hardware impact to do this function is probably less than 1.0 kg and 1.0 watt. A simple change to the TT&C subsystem would be to add a command sequencer activated by a command decoder output when no command had been processed within some selected time interval. This function could also be implemented easily in the RMS approach.

4.6 AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF THE S/C PROPULSION FUNCTION*

Maintenance of propulsion function health (see hierarchy in Figure 4-25) requires extensive analysis by the propulsion function using sensor and spacecraft reaction data. Input is required from attitude control and navigation. The output is used in decision making by the propulsion, attitude control, navigation and system functions. An example of an on-board algorithm for thruster fault protection was presented in Section 4.3.5.3.2 as it might be implemented in the existing ACS computer.

4.6.1 Maintain Thruster Health - Level 2/Category I

Analysis requires both thruster data from sensors and reconstruction of thruster performance as inferred from attitude control and navigation data. Direct thruster data is currently limited to catalyst bed temperature but additional sensors (such as chamber pressure) could augment on-board monitoring of performance. Software additions to cross-check pulse performance during reaction wheel unloading, and steady state performance during navigation maneuvers can also be implemented on-board.

To provide additional flexibility and increased depth of fault protection, consideration should be given to isolating thrusters in smaller blocks (e.g., pairs) rather than in two branches.

4.6.2 Maintain Propellant System Health - Level 2/Category I

Analysis for propellant leakage can be performed during quiescent periods by monitoring existing tank pressure and temperature sensors with on-board software. During periods of thruster activity, estimates of propellant usage can be obtained using the same software required to support attitude control and navigation functions. Because of the limited sensitivity of tank pressure as an indication of leakage, as much cross-checking as possible should be incorporated to detect excessive use or leakage as soon as possible.

*By R. W. Rowley

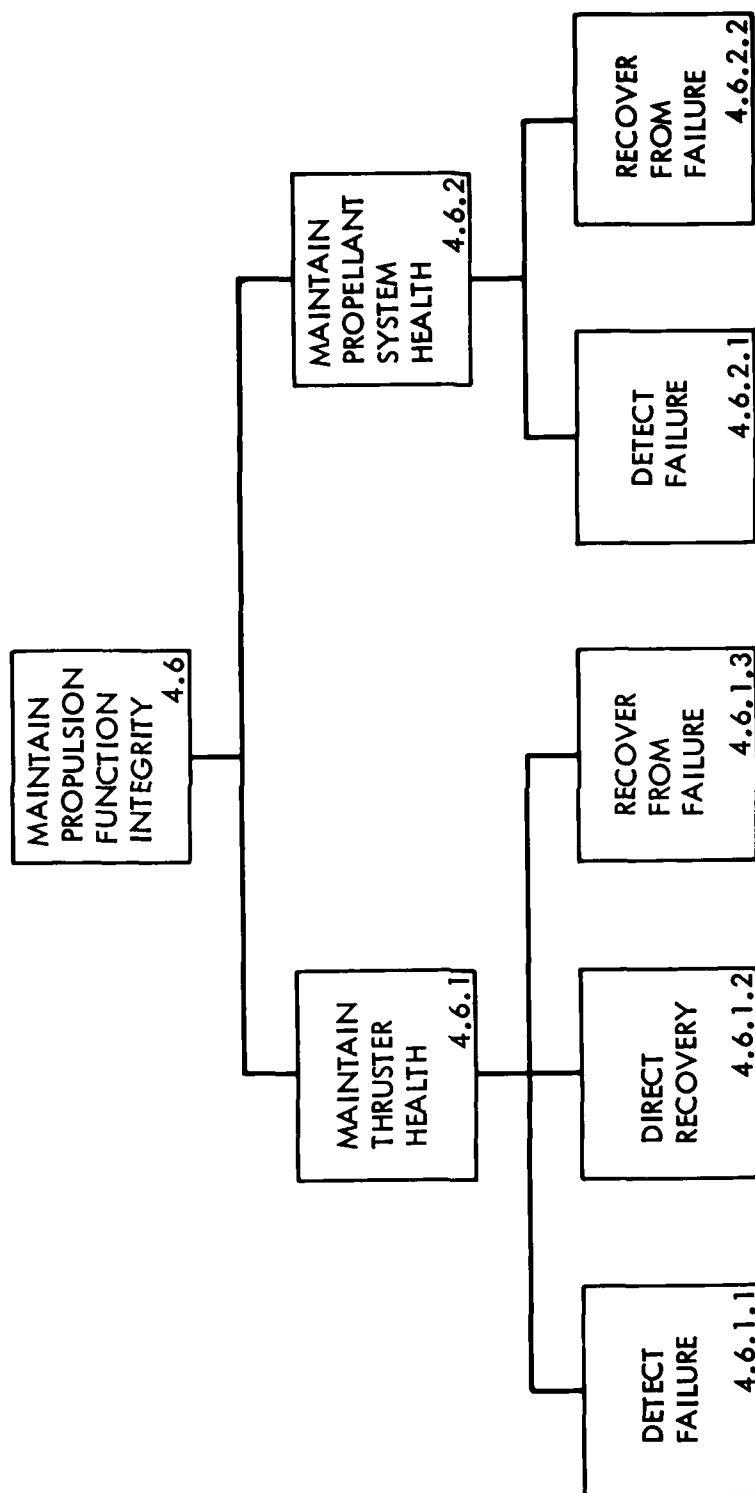


Figure 4-25. Propulsion Integrity Maintenance Functional Hierarchy

4.7 AUTONOMOUS OPTIONS TO MAINTAIN INTEGRITY OF STATIONKEEPING FUNCTION*

This section discusses autonomous maintenance of the tracking function, and autonomous maintenance of the autonomous stationkeeping function. Figure 4-26 is a hierarchy of the autonomous stationkeeping maintenance function.

4.7.1 Maintain Tracking Function - Category II

The tracking service in 2.7.1 of Volume II, "Provide Tracking Function," provides the means for spacecraft orbit determination through the use of the S-Band RF uplink and downlink signals. Basically, the maintenance of this service is a redundancy management function directly associated with the maintenance of the uplink and downlink signals (i.e., 4.5.1 "Maintain Telemetry function" and 4.5.2 "Maintain Command Function"). Even if the determination of spacecraft orbital position is automated on-board the spacecraft (2.7.1) it may be desirable to have ground tracking as a back-up function. If so, autonomous maintenance of the on-board capability for performing the tracking function would be necessary.

The current autonomy features of the tracking maintenance function, which are minimal, are the same as those discussed under the command and telemetry sections. Also, active ground participation is required to command the ranging channel on and off and to command tracking in the coherent and non-coherent mode. These latter two commands are associated with the tracking function.

To provide the redundancy management required to maintain the tracking function the additions or options discussed under Maintain Telemetry and Maintain Command apply for tracking. The additions in the command and telemetry areas are somewhat one to one (i.e., the Telemetry Failure Sensing scheme would be considered for command and telemetry maintenance and thus for tracking maintenance).

In addition to the command and telemetry options above there are a few other items which might be done for tracking maintenance.

4.7.1.1 Leave Ranging Channel On. The ranging channel could be left on all the time. In this way it is available immediately. However, since an uplink needs to be established for tracking, the channel could be ground commanded on.

4.7.1.2 Provide Cyclic Ranging Channel. If for some reason it was desired to have ranging channel go on and off without ground commands, there are ways that this could be implemented automatically (e.g., a cyclic on/off, a ranging signal detector, or an AGC switch ranging on/off).

*By S. O. Burks, J. R. Jones and P. R. Turner

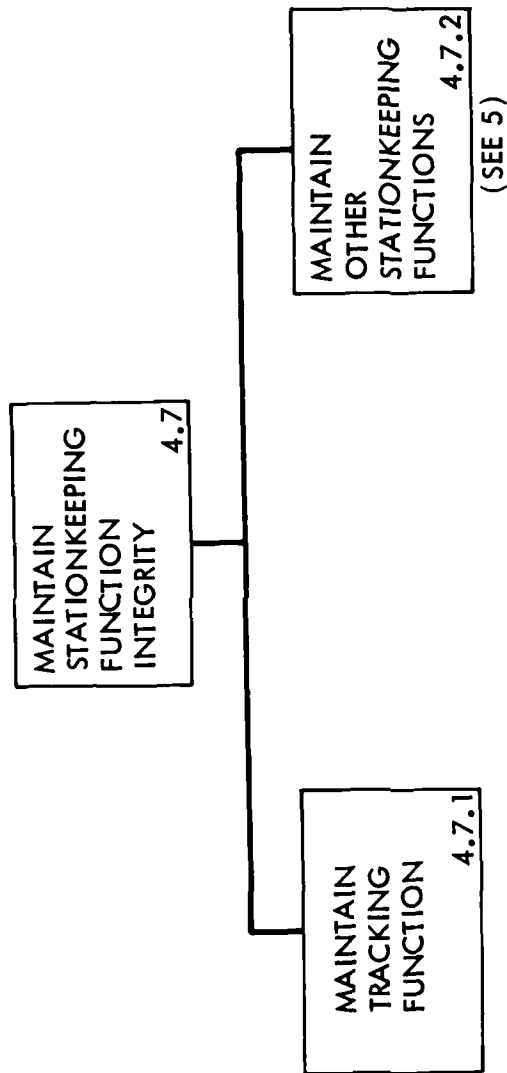


Figure 4-26. Stationkeeping Integrity Maintenance Functional Hierarchy

4.7.1.3 Provide Non-Coherent Ranging Mode. In the DSCS III design implementation any S-Band redundancy management scheme could result in the selection of redundant elements which could eliminate two-way coherent tracking. A receiver A/transmitter B or receiver B/transmitter A combination eliminates two-way coherent tracking (i.e., the coherent drives are not cross-strapped from receiver to transmitters). The ranging modulation is cross-strapped so there is a non-coherent ranging mode which is possible but is not normally used.

Any autonomous redundant element selection scheme should attempt to maintain the tracking function as well as a telemetry or a command function (e.g., if receiver A failed both receiver A and transmitter A should be switched to their block redundant "B" counterpart).

4.7.1.4 Leave in Coherent Mode. Ground command currently can select a "non-coherent" mode when the receiver is in lock. For autonomy the system should be left in the coherent mode (as it normally is). The coherent/non-coherent mode selection could be handled similarly to the ranging on/off function in 4.7.1.2 above.

4.7.1.5 Provide Redundant Ultrastable Oscillators. The use of redundant Ultra Stable Oscillators on the S-Band downlink-only signal might provide sufficient stability for one way tracking. However, this would require adding a stable oscillator or synthesizing the correct frequency from the comm subsystem frequency reference (stable oscillators). To maintain this approach would require the automatic options addressed under the telemetry section in Section 4.5.

4.7.1.6 Use X-Band for Tracking. It might be possible for the X-Band link to be used for tracking. The comm subsystem frequency reference might be stable enough for required tracking accuracies. However, for the ground stations to use this signal, process the data, and compute orbit vectors requires major changes to the DSCS III ground stations. The maintenance of this function would be the same as that suggested under the telemetry section for X-Band. The spacecraft impact is minimal for this approach but the ground impact could be very significant.

If the payload users have to know where the spacecraft is and they have to listen to the X-band signal for payload information then the X-Band signal may be a readily available source of orbit data.

4.7.2 Maintain Autonomous Stationkeeping Function - Category I

This section assumes that the autonomous navigation function whose characteristics are described in Section 2.7, is implemented. If so, the system itself must be fault tolerant to ensure that the spacecraft autonomy requirements are met.

Given a baseline on-board autonomous navigation system, which provides the normal (no failure) operation of the spacecraft, an upgrade to support a fault-tolerant design requires the addition of two categories of functions. First, the navigation system must monitor its operation and performance and detect and correct failures within the on-board navigation system. Secondly, the navigation system will be required to generate data to support fault detection in other spacecraft systems.

4.7.2.1 Fault Detection. Since the navigation system contains both hardware and software, the first category of upgrade involves a number of failure detection techniques. For example, to detect sensor failures a number of redundant sensors may be employed. Comparison of their outputs with both their redundant companions and with the expected values from the navigation process will provide both failure detection capability and switching logic. Internal software failures must also be detected. While software failure detection is much more difficult, potential techniques include:

- (1) Precedence checks to ensure that functions or tasks that must be performed in a sequential order are not executed until the successful completion of their predecessors.
- (2) Magnitude limit checks to ensure that computation of a critical, well-defined quantity has not produced a value outside its limits.
- (3) Sign checks may be accomplished to verify that an output quantity has a sign compatible with inputs used in its calculation.
- (4) Invariant constants may be evaluated to insure that the quantities used to calculate them are physically consistent.
- (5) Filtering processes may be applied to reduce the effects of statistical errors in data and to make the navigation process tolerant of intermittent or transient faults in data.

4.7.2.2 Support Functions. Support of the autonomous features of the other spacecraft systems will, in general, require the navigation system to compute quantities not directly required by the navigation process. Examples of these extended requirements include:

- (1) Prediction of lunar and solar occultation periods in support of the attitude control system.
- (2) Reconstruction of spacecraft V maneuvers in support of thruster failure detection and attitude control performance monitoring.

SECTION 5

VALIDATION AND OPERATIONS ISSUES FOR AUTONOMOUS DSCS III SATELLITES*

Autonomous spacecraft require validation of their autonomous functions during design, assembly, and test, and during flight operations. In addition, the characteristics of ground operations of spacecraft with autonomy will be quite different than for the current DSCS III. This section discusses some issues related to validation and operation of an autonomous DSCS III spacecraft.

5.1 PHILOSOPHY

A primary goal of autonomy is to reduce dependence on ground stations. The degree to which dependence is reduced is a function of:

- (1) The degree to which the satellite is autonomous, and
- (2) The degree to which the ground system is willing to trust the autonomous features.

The first item results in a trade between ground and spacecraft validation costs vs ground operations costs. That is, the more confident the operators are in the reliability of the spacecraft autonomy the smaller can be the operations team to monitor the autonomous features. Issues of security, reliability, and readiness must be considered in making these cost trades. The following discussion deals with some issues in making these trades.

5.1.1 Validation Philosophy

Validation philosophies and requirements must be developed early in the Project process. As the spacecraft features approach that of a Level 5 autonomous design it becomes evident that all system functions, performance characteristics, and mission modes can no longer be tested on an individual subsystem or system basis. Systems and subsystems which include redundant elements, complex interfaces, autonomous functions and/or intricate mission sequence configurations, require an integrated, end-to-end validation program structure. Testing must begin at the lowest reasonable level and a 'building block' approach must be used where each test complements those tests to follow. Early development phase engineering interface tests must be scheduled for complex autonomous features. Special hardware and software must be provided to execute test sequences. In addition, interface diagnostics must be available to isolate failures to the replaceable spares level.

*By S. O. Burks, R. C. Detwiler, R. Malm, E. Mettler, R. N. Miyake, B. L. Sharpe, and P. R. Turner

5.1.2 Operations Philosophy

The Air Force Satellite Control Facility (AFSCF) can be broadly characterized as being comprised of:

- (1) Physical facilities
- (2) Computer hardware, software, and ancillary devices
- (3) Personnel
- (4) Documentation: directive, procedural, descriptive, etc.

Each of these resource categories, and the schedule which dictates the total amount and frequency of their use, can be affected by the configuration of the autonomously-operating satellites ultimately selected. Conceivably the total ground effort required to control these missions could increase, decrease, or remain about the same as that required for present (non-autonomous) DSCS III satellites, depending on some basic assumptions chosen. The factors or issues governing these assumptions are:

- (1) The allowable amount of additions or changes to the SCF to support autonomous satellites.
- (2) The allowable increases to scheduling real-time and non-real-time support activities (multiple DSCS and multiple mission considerations).
- (3) Required additions to procedures and tools used to monitor and control autonomous satellite systems.
- (4) Extent of usable backup capability to monitor and control using DCA's SHF command and telemetry links.
- (5) Potential tradeoffs in specified limits of stationkeeping, system performance, etc.

A mission operations (ground) activity profile can be considered for three mission phases:

- (1) Satellite on-orbit checkout phase
- (2) Ground assisted operations checkout phase
- (3) Autonomous operations phase.

Conceivably, a spacecraft configuration could be chosen which would require a constant, high level of ground monitoring and control activity in order to be maintained in an "autonomous-ready" condition. This configuration might require daily updating of a large number of values stored on-board in order to assure that a probable 6-month command-free period could begin on any given day. This scenario seems unbalanced, and probably would not provide reasonable time or resource margins in the event of failures occurring in the spacecraft or on the ground, particularly if the effort were being divided between several on-going missions.

Likewise, it is possible to envision an opposite limiting case - an autonomous system so advanced and sophisticated that after its on-orbit checkout it would require no further interaction with ground control. This type of ultimate autonomous satellite seems almost equally unlikely.

A reasonable compromise for a baseline activity profile would be one resembling that presently used for non-autonomous DSCS III; that is, a high level of support during on-orbit checkout, decreasing to no more than 12 to 24 contacts per year thereafter for each satellite. Although subsystems would be functioning autonomously during the "ground assisted operations" phase, the ground would still be required to predict and verify many of these autonomous actions. Presumably the increase in level of effort in monitoring autonomous subsystems could be designed to balance the decrease in effort required to control many of the operations presently controlled from the ground. Savings in operations costs due to autonomy can only be achieved by careful tradeoffs between spacecraft capability, sophistication of the validation process, and the requirements for ground monitoring of the spacecraft.

In functions which require large numbers of sequential ground commands, on board sequencing can be a substantial contributor to 6 month operations without ground intervention. Sequencing is suitable for activities which can be predicted, and for which no on board decisions are required. Sequencing is most applicable to routine services functions. For example, on DSCS III, initialization sequences are stored in the ACS. On board sequencing is thus regarded as an autonomous feature. On board sequencing substitutes pre-event ground activities for real time activities. Sequences must be generated, validated, and transmitted in advance. However, real time operations can then be devoted to unexpected events (such as failures) or events with unpredictable characteristics (such as maneuvers). Sequencing trades off the chance of sequence design and validation errors and of on board faults affecting the sequence, for the chances of errors in individual command generation and transmission.

5.2 VALIDATION/OPERATIONS METHODS

5.2.1 Spacecraft State Simulation and Analysis

One aspect of ground control and validation of the DSCS III autonomous spacecraft which will deserve some ongoing consideration is the question of necessity for a spacecraft simulator or emulator on the ground, used to predict or reconstruct autonomous activity. JPL missions, notably Viking and Voyager most recently, have employed models of their on-board computer systems in order to predict and test both algorithm coding instructions and sequences to be executed. (Modeling of physical environment and system behavior has not been used.)

Hardware or software simulators are required to provide test sequences and to thoroughly investigate anomalies. Simulators may be provided by software, breadboard units, or prototype hardware upgraded to flight configuration and delivered for test use. This upgraded hardware set, spare subsystem element, or a hybrid software/hardware simulator would then be assembled in a test bed configuration with the current flight software. During test activities (and subsequent flight operations) problems are investigated using the test bed in parallel with continuing operations.

It should be noted that test bed hardware, operated and maintained by the flight organization on an as-needed basis, is often a more cost effective validation tool than a detailed software simulator. In addition, test and training activities are enhanced by utilizing actual flight system elements. The test bed also acts as a project memory thus permitting an orderly transition of personnel to other tasks without the loss of key spacecraft performance information.

Another issue to be addressed in more detail during design is the nature and extent of the data base required to track and describe specific functions operational characteristics in a multi-spacecraft environment. As the spacecraft lifetime increases, failures and other operational idiosyncracies may cause each spacecraft to diverge from the design baseline to a significant enough degree to require separate data bases for the control of each. These data bases may take the form of either documentation or ground software. The extent of the effort required to maintain these data bases sufficiently current and accurate is not known.

5.2.2 On-Orbit Verification of Autonomous Operations

On-orbit verification of proper autonomous operation requires:

- (1) Initial generic testing of the first-launched satellite to acquire detailed operating characteristics and a good baseline of experience. Each autonomous function must be tested in various configurations, more than once.
- (2) Initial test and checkout of each subsequent satellite, wherein each autonomous feature is checked to give some baseline level of confidence.

(3) Periodic verification of proper execution of a specific function, e.g., a battery charging or stationkeeping maneuver.

(4) Periodic verification of redundant elements.

To keep the ground operations activity at a reasonable level it seems prudent to conduct testing as in (1) above only once, and testing as in (2) above with only one spacecraft at a time. It may be desirable to test as in (2) periodically -e.g., after autonomous mission phases, or yearly. Testing as in (3) and (4) would be routinely done, and would fall into the same category as frequent health and status checks.

The major requirements and performance characteristics which dominate the validation process are:

(1) The Level 5 autonomous design option and the requirement for transparency of autonomous features to mission users

(2) The 60 day/6 month autonomous operations requirement

Each of these design requirements will impact validation philosophies, methods, schedules, hardware and software, and operations practices.

5.2.2.1 Level 5 Autonomous Design Option and Autonomous Feature Transparency. Self-test of critical hardware and software functions is essential for the Level 5 class spacecraft with autonomous action transparency. Self-test includes, as a minimum, tests of memory, command execution coding, and external interfaces. More sophisticated self-test capabilities would help reduce periodic maintenance requirements.

Software designs should be structured with a 'table driven' architecture to easily accommodate criteria, format, and parameter changes without re-validation of in-line code.

Experience has shown that a significant number of detected failures are the result of limited understanding and documentation of system performance characteristics. A test bed and an independent anomaly operations group must be provided to allow investigations to be conducted with minimum effect to continuing test or flight operations.

5.2.2.2 60 Day/6 Month Operations Requirement. Scheduled test activities must include a 60 day test period with minimum ground intervention and minimum ground support equipment. This 60 day test period could be included as part of the system acceptance environmental tests or during a flight operations test and training program.

For this test period special cruise mode software program should be developed which cycles all space element modes, data rates, interfaces and spares to allow ground elements to verify proper end-to-end system performance. Validation of the 6 month survivability requirement would include an analytical model in combination with an engineering test soon after

orbit insertion. This early engineering test would be used to confirm the results of the analytical model algorithms and to provide data to update spacecraft parameter tables.

'Safe mode' software might also be provided in protected ROM to ensure survivability under transient conditions and operational configurations which were overlooked in the validation program. During on-station validation of the autonomy features, the protected ROM software would provide a degraded, safe configuration until ground communications could re-establish nominal performance.

5.2.3 Impacts of Spacecraft Autonomy on Payload Control and Utilization

The DSCS III payload will require the same functions from an autonomous spacecraft as from the existing spacecraft. Some of these functions impact the DCA "users" more or less directly. These include the Spacecraft Control and Monitoring Function and the Stationkeeping/Navigation Function. Most of these issues will arise only if the spacecraft control and tracking functions on the ground are completely eliminated.

5.2.3.1 Payload Control Issues with Regard to Control, Monitoring and Tracking. Discussion of current practice is contained in Volume II, Section 5.2.

5.2.3.1.1 Payload Reconfiguration. The autonomous DSCS III TT&C subsystem will certainly continue to process network reconfiguration commands for the payload through the SHF link. It is not clear whether payload redundant element switching will still be accomplished through the S-Band link, or whether it will be changed to the SHF link. If the payload redundancy management were made autonomous, the payload control ground system load would be alleviated.

5.2.3.1.2 Timing. The 5 MHz frequency standard is provided to the users through the SHF TT&C beacons. Its frequency is updated by commands sent over the S-Band link. If the spacecraft is operating autonomously the frequency standard updates might have to be generated and their accuracy maintained by the payload control operations over the X-Band TT&C command link. If S-Band turn-around ranging and Doppler measurements are not continued during autonomous operation then a different form of reference for frequency standard accuracy must be maintained.

5.2.3.1.3 Payload Integrity Maintenance. Initially it will probably not be practical to implement total spacecraft autonomy. Ground control will continue to be necessary or desirable for certain integrity maintenance functions. It is not clear whether this should be accomplished through the SHF TT&C link because it will also be there for payload reconfiguration, or whether the S-Band link should remain primary for all spacecraft-ground interactions other than network-payload reconfigurations.

5.2.3.1.4 Ephemeris Information. Certain comm system users will require accurate ephemeris information. This in turn will require that orbital state vectors be transmitted from the autonomous navigation system on board the spacecraft. Payload ground operations will need to be established to receive and utilize this vector. Decisions will also be required as to whether the vector will be transmitted over S or X-Band, or both.

5.2.3.1.5 SHF Command Failure. The X-Band command link cannot withstand some single point failures without the S-Band command link being used to activate some of the redundant elements (e.g., the KI-24 decryptor or the comm system frequency reference). Some method of resolving this problem must be developed. Possibly the design could be modified so that the redundant KI-24 can be left on and a totally redundant SHF command channel is available with only a ground frequency change. Perhaps the appropriate failure sensing can be designed and the autonomous TT&C redundancy management function could provide fault protection without ground intervention.

5.2.3.1.6 Command/Telemetry Redundancy. The technical capability for performing all telemetry and command functions exists at either SHF or S-Band. However, payload ground operational changes would have to be implemented to make any alterations in current standard procedure. Some of these may have far reaching impacts, and this topic will require further investigation.

Obviously, the more autonomous the spacecraft becomes in managing faults, the less the payload operations units will have to do beyond controlling the network configuration. For example, if the payload redundancy management is made autonomous along with the other spacecraft systems, the payload control ground system load would be significantly alleviated.

5.2.3.2 Options for Payload Control Substitution for Mission Control. A question has been briefly addressed regarding how much support an autonomous spacecraft might be supplied by DCA conducting monitoring and control of the spacecraft via the SHF (X-Band) links, presumably in the absence of AFSCF. If the spacecraft can be "helped" through some of its activity by any of the (fixed or mobile) DCA elements, this may be a reasonable method of trading off some of the functions which are difficult to implement on board the satellite or to phase in autonomy.

Since the communication link must be maintained for control of the DSCS III communications payload, these links could conceivably be used to control some spacecraft functions. Such control could be used in lieu of providing on-board autonomy for very difficult-to-implement functions. Also, payload control could assume some spacecraft control functions during the period of phasing to a completely autonomous spacecraft. This would allow phaseout of overseas spacecraft control stations (for example) sooner than would otherwise be possible.

5.3 FUNCTIONS

VALIDATION/OPERATIONS IMPACTS AND ISSUES FOR SOME AUTONOMOUS

5.3.1 Issues in Validating Operation of the Autonomous Power Function

5.3.1.1 Battery Charging. Autonomous battery charging might be validated in space during eclipse season by judiciously turning off the battery charger to partially discharge a battery while observing the automatic selection of battery charging parameters. Another validation option is to perturb battery charging parameters (e.g., select low charge rate and lowest charging current V-I characteristic) by ground command while again observing the autonomous response.

5.3.1.2 Load Management (e.g., Load Control to Maintain Battery Energy). The existing design requires extensive ground segment analysis and control of load power in the management of stored energy and battery life. With autonomy, direction of load management would be transferred to the space segment. Validation after launch is not recommended because it results in temporary loss of low priority loads if load shedding occurs.

5.3.1.3 Battery Reconditioning. The batteries in DSCS III may not need reconditioning for several years, so it is probably unnecessary to provide autonomous battery reconditioning for this mission to meet the desired autonomy goals. A simpler approach would be to initiate the reconditioning sequence by ground command early in a period of no eclipses and have the spacecraft automatically complete the sequence and restore the battery to normal use. Ground operations in this case consist of analysis to determine the need for reconditioning, and issuing a command to initiate the reconditioning sequence. In-orbit validation can be inferred by observing battery parameters after a reconditioning command has been issued.

On the other hand, if the spacecraft is sophisticated enough to accurately determine and control battery charge status, it is probably a small additional change to fully automate battery reconditioning. Validation in this case would be difficult in flight, but could be checked before launch by injecting dummy signals into the battery sense inputs.

5.3.1.4 Redundant Load and Converter Selection. Without the benefit of ground based health checks on various subsystems it is almost impossible, with the existing design, to determine whether an anomaly has originated in the dc-dc converter or in a converter load. With additional current and voltage sensors on each converter output and input, a computer on the spacecraft could autonomously distinguish between a converter fault and a load or source fault, and subsequently direct switching to the appropriate redundant block

Validation of autonomous load and converter selection would be tested by injecting dummy signals into the converter voltage and current sensors prior to launch and observing the spacecraft computer response.

A simpler alternative that does not require numerous additional sensors is to switch all redundant loads when an anomaly is perceived. If the anomaly persists then the redundant converter should also be substituted. The disadvantage of this approach is in failing to identify the source of anomaly, but further diagnostic switching, if desirable, could be employed for that purpose during noncritical mission phases.

5.3.1.5 Operations with a Failed Battery (Battery Chain Failures). The existing design requires ground segment analysis and control of battery charge rates, battery heaters and the Battery Charge Regulator to determine that there is a battery failure and remove it from the bus. With an autonomous system, battery failure analysis and switching would be transferred to the space segment and the ground segment would be utilized only to check and verify space segment decisions, and provide an override/reprogram function, should it be required. Once a failed battery or batteries have been removed from the bus, the autonomous load management function and/or special subsets of this function will be required to maintain battery energy, keep peak loads below the solar array capability, and power down the spacecraft prior to eclipse.

5.3.2 Issues in Validating/Operating the Autonomous Attitude Control Functions

Three options exist for ACS validation/verification/test (all are the same in this context) during routine mission operations:

- (1) No further validation after the initial on-orbit check-out, unless ground-directed due to a real fault event.
- (2) Infrequent dedicated test periods which assume user interference by exercise of the autonomy functions, similar to the initial check-out. This represents a compromise strategy with respect to (1) and (3).
- (3) True autonomy validation, or end-to-end self-test, in a non-interference manner, performed at frequent, regular intervals to ensure a continuously high probability of extended-time autonomous capability. (This represents the ultimate goal.)

The remainder of this discussion will cover the feasibility of Option 3.

Validation of autonomous functions in the actual mission, without disturbing the payload user, may be limited to indirect measurement and inferences rather than true input/output response tests. The chief problem is the constraint of maintaining user-normal spacecraft motion and orientation states. This involves the use of pseudo or synthetic sensor stimulus, actual response, and spacecraft state change to obtain an equivalent end-to-end test. There is a difference between built-in testing, which can allow the on-line equipment to self-test via health checks and message validation and true autonomy self-test. This is because of the need to test the integrity

maintenance functions, particularly redundancy switching control and hand-off to spare units. In other words, the front end of the fault tolerant process (error detection) is more readily tested by pseudo-fault injection/detection techniques than the final process of recovery, which calls for various reconfiguration levels. The driver is still the user transparency requirement.

5.3.3 Options for Operating an Autonomous Thermal Control Function

The current thermal control subsystem (TCS) ground command function is limited to an on/off command of the control heater system. Once the control heater system is enabled, the thermal control subsystem operates autonomously. Full autonomy would require that the control heater system be enabled at all times with a system to disable the control heater system if or when a survival condition is encountered by the spacecraft. The primary heater system should be disabled for survival mode conditions. For the TCS the survival mode occurs when the vehicle must be powered down to conserve electrical power. Disabling operational heaters conserves power because survival heaters set points are at a lower temperature. Therefore, the ground ops would shift from routine "enable" control to a "disable" override control for abnormal situations.

5.3.4 Ground Operations Considerations for an Autonomous Spacecraft Control and Monitoring Function

5.3.4.1 Telemetry. The possible ground impacts of implementing various autonomy options to detect and correct for telemetry failures are:

5.3.4.1.1 On/Off Sequencer. For this option the S/C and S-Band downlink telemetry is on almost continuously (the X-Band downlink Beacons are normally on continuously). A sequencer is used to turn the X and S-Band telemetry function on and off for a variable time and duration, as required for S/C operations.

The ground station's design would not change. The basic S/C ground operations at X-Band, and more so at S-Band would change somewhat to accommodate a new S/C mode. All TT&C failure detection correction would be accomplished on the ground, and would typically be a selection of redundant elements.

5.3.4.1.2 RF Power Monitor. For this option the spacecraft will automatically switch to redundant units if an RF level detector senses low RF output power from the S-band or the X-Band transmitters.

The ground station design would not have to change. The operations with spacecraft would have to change slightly to accommodate a new mode (for example, switching to the redundant channel frequency or synchronizing the telemetry if there was an automatic switch of the S-Band transmitter). There are several failure modes of the telemetry function which

still would require ground detection and correction by selection of redundant units.

5.3.4.1.3 Telemetry Failure Sensing. For this option the S/C TT&C subsystem health or status is assessed by a "Redundancy Management Subsystem" which samples the S/C TT&C engineering telemetry data stream. If a failure were "sensed", redundant elements would be switched to correct the failure.

The ground station design would not have to change. The operations with the S/C would have to change slightly to accommodate a new S/C operating mode. For example, the ground station might have to re-acquire the downlink carrier at X-Band, if the communication subsystem frequency reference failed and was switched. "Telemetry sensing" could correct several types of failure modes. Several failure modes of the telemetry function would still require ground detection and correction by command selection of redundant units.

5.3.4.1.4 Direct Failure Sensing. For this option the S/C performs a self-check of the telemetry function by sampling the transmitted RF signal and checking the quality of the signal. If a failure is detected, redundant elements are selected to correct the defect. The ground station design would not have to change for this option. A new S/C design implementation would require a slight modification to the planned operations with the S/C. This direct sensing technique should correct for almost all failure modes. Ground failure detection and correction efforts should be minimized with this approach.

5.3.4.2 Command. Making the command channel autonomous basically removes the ground decision to change the frequency and to do some failure sensing and correction. The major change in the ground segment by using the options below is the reduction of the ground segment decision to change frequencies, and a few other minor decisions.

5.3.4.2.1 Use As Is. For this option of using the DSCS III design as is, no significant changes are required to the ground segment. The current DSCS III S-Band uplink is somewhat autonomous. It does require the ground to select a frequency for transmission and/or a command word preamble if the first attempt to command over the S-Band link does not work. Also, for specific failure modes, the X-Band command channel is available in a similar manner.

5.3.4.2.2 S-Band Frequencies. This option converts the redundant S/C command channels to the same operating frequency. This allows immediate access to either redundant channel without a ground transmitter frequency change.

There should be no impact on the ground station design, but the planned operating procedures will have to be modified to account for a new command acquisition strategy.

5.3.4.2.3 X-Band Dual Carrier Transmission. For this option the X-Band ground station would transmit simultaneously, both of the redundant X-Band command channel frequencies (i.e., channels 1 and 5). This would allow, immediate access to either redundant command path.

This option is not recommended. There are several potential problems such as ground receiver interference (intermodulation products) and uplink interference. These types of problems could take a great deal of design effort. The station probably would have to be redesigned to have simultaneous dual carrier transmissions.

5.3.4.2.4 Telemetry Failure Sensing. This option could use a "Redundancy Management Sub-System" (RMS) approach of the engineering telemetry stream. The TT&C data is reviewed by the spacecraft RMS for health and status of S and X-Band command links, and the S/C takes appropriate autonomous actions based on the data.

The ground station design should not have to change for this approach, but the operations procedures would change slightly. Ground operations would still have to monitor this correction activity. However, some failures might not be sensed via the telemetry stream. These failures would require correction by ground commanding of redundant elements.

5.3.4.2.5 Direct Command Function Failure Sensing. This option uses an on-board command transmitter to test the command channel. The ground station design might have to change depending on the test command implementation. This test command has to be overridable from the ground (even with a failure) and this condition might require a change.

5.3.4.2.6 Cyclic Command Not to Switch. This option uses a ground transmitted command to validate the command channel on a periodic basis. If a ground command is received and processed within a certain period of time no action is taken. If no command is processed within a certain amount of time a spacecraft failure routine switches redundant elements until a valid command processed.

This approach should not affect the ground station design. However, the operations will have to be modified to assure a command periodically" and to accommodate a new design approach. This should be a relatively moderate change.

5.3.4.3 Tracking. The ground impacts due to command and telemetry options addressed above apply to S-Band tracking. There are some additional possible minor changes due to the tracking function autonomy, which include the method of using the ranging channel and the "non-coherent/coherent mode". these changes will have a minor effort on ground operations and no effect on the ground design. There are, however, some ramifications of using the X-Band system for tracking and on-board orbit determination, and involving the DCA EDM-SCCE. These were discussed in Section 5.2.3.2.

5.3.5 Autonomous Navigation Considerations for Mission Operations

The ground systems associated with the autonomous navigation subsystem will support executive override of the on board subsystem, validation of proper operation, and updating of on-board software and data base values.

5.3.5.1 Executive Override. The current DSCS Control Program (DCP) and associated software described in Reference 3 will suffice to allow ground override of the on board navigation subsystem. The ground system will operate in the standard non-autonomous DSCS III mode to allow for orbit determination, ephemeris maintenance, and maneuver planning/command generation as an alternative to autonomous on-board navigation.

5.3.5.2 Validation of Navigation Subsystem Operation. A series of ground functions are required to validate the proper operation of the on-board navigation subsystem. These functions are also required in the basic ground testing of the on-board subsystem and may be derived from the hardware/software used in validation testing prior to flight.

5.3.5.2.1 Telemetry Downlink Requirements. The ground system must be capable of processing the following types of navigation subsystem data and storing for failure display and analysis:

- (1) Navigation subsystem memory contents, including software and data.
- (2) Navigation subsystem audit trace data produced for validation of autonomous operation.
- (3) Any special fault detection/correction data not associated with (2).

5.3.5.2.2 Validation Analysis. The ground system must be capable of analyzing downlink information to determine that the information contained in the navigation subsystem audit trail reflects a properly functioning subsystem. The analysis capability should allow for assessment of the navigation strategy effectiveness and the proper performance of each function in the navigation subsystem.

5.3.5.2.3 Software Modification and Update. The ground system must allow revised versions of the navigation subsystem software to be loaded for correction of design errors, improved response to existing fault occurrences, and provision of improved performance. The system will support validation of the software modifications with an appropriate degree of simulation or emulation of the navigation subsystem. The ground subsystem will provide for formatting of software loads for uplinking to the on-board subsystem.

5.3.5.2.4 Data Base Update. The ground system must provide the capability to modify the values of on-board data consistent with a baseline software load or an updated software load. Proper formatting of data loads for uplink will be provided. Such data updates may support the autonomous operation by providing periodic updates to star catalogues, luni-solar ephemeris data, or other time varying models that may be included in the navigation subsystem design. This will also support the "tuning" of the autonomous operation with data for specific station locations or changes in performance due to faults or normal spacecraft aging.

REFERENCES

1. Marshall, M. H., "Goals for Air Force Autonomous Spacecraft," JPL Internal Document 7030-1, Preliminary Issue, 2 February 1981.
2. Informal Communication, C. H. Bredall, Aerospace Corporation, to D. D. Evans.
3. DSCS III Orbit Operations Handbook, CDRL-A049, November 1980.

APPENDIX A
LEVELS OF AUTONOMY

(Reproduced directly from Reference 1)

APPENDIX A

LEVELS OF AUTONOMY

(Reproduced directly from Reference 1)

In performance of a space mission, four major policy goal categories have been identified. These are:

- (1) Ground interaction reduction.
- (2) Spacecraft integrity maintenance.
- (3) Autonomous features transparency.
- (4) On-board resource management.

The extent to which these goals have been accomplished to date has been through a mix of functions resident in either the space segment or the ground segment. Furthermore, the ground segment, as an integral part of the total system, has been responsible for accomplishing maintenance, navigation mission control, and payload data processing. Thus, only minimal spacecraft autonomy has been needed.

The levels of autonomy described in this appendix are used to define a step-wise increase in spacecraft autonomous capability. By proceeding through the levels, autonomous capability is increased in the space segment and dependency on the ground segment is reduced.

The levels of autonomy are described as follows:

Level 0. A design without redundant elements which meets all mission needs by operating without the on-board control of state parameters (such as rates and position). May respond to a prespecified vocabulary of external commands, but cannot store command sequences for future time- or event-dependent execution or validate external commands. (An open-loop, on-board system controlled from the ground.)

Level 1. Includes Level 0 but uses on-board devices to sense and control state parameters (such as rates and positions) in order to meet performance needs. Is capable of storing and executing a prespecified command sequence based on mission-critical time tags. Will respond to prespecified external commands, but cannot validate external commands. Functionally redundant modes may be available for a degraded-performance mission.

Level 2. Include Level 1 plus the use of block redundancy. Ground-controlled switching of spare resources is required. Uses cross-strapping techniques to minimize effect of critical command link (uplink) failure modes. Significant ground-operator interaction is required to restore operations after most faults if spare spacecraft resources are available.

Requires operator interaction for fault recovery. Is capable of storing and executing mission-critical events which are sensed on-board and may be independent of time.

Level 3. Includes Level 2 and is capable of sensing prespecified mission-critical fault conditions and performing predefined self-preserving (entering a safe-hold state) switching actions. Is capable of storing contingency or redundant software programs and being restored to normal performance (maintaining the command link with a single link fault) in the event of a failure. Timers may be used to protect resources. Requires ground operator interaction for fault recovery. In general, the failure to sense and/or execute the mission-critical event(s) will cause mission failure or loss of a major mission objective.

Level 4. Includes Level 3 but is also capable of executing prespecified and stored command sequences based on timing and/or sensing of mission events. Ground-initiated changes to command sequences may be checked on-board for syntactical errors (parity, sign, logic, time). Uses coding or other self-checking techniques to minimize the effects of internally generated data contamination for prespecified data transfers. Requires ground-operator interaction for fault recovery. In general, failure to sense and/or execute the mission event(s) or state-changes (excluding failure-induced state-changes) will cause mission failure or loss of a major mission objective.

Level 5. Includes Level 4 and is also autonomously fault-tolerant. Is capable of operating in the presence of faults specified a-priori by employing spare system resources, if available, or will maximize mission performance based upon available capability and/or available expendables (i.e., self-loading of contingency programs) without ground intervention.

Level 6. Includes Level 5 and is capable of functional commanding with on-board command-sequence generation and validation prior to execution. Functional commanding may include a high-level, pseudo-English language, spacecraft-system/operator communication and control capability.

Level 7. Includes Level 6 and is capable of autonomously responding to a changing external environment, defined a-priori, so as to preserve mission capability. The capability to change orbit in order to compensate for degradation or to protect the satellite from an external threat is included.

Level 8. Includes Level 7 and is capable of operating successfully within the presence of latent design errors which could cause loss of major mission objectives.

Level 9. Includes Level 8 and is capable of task deduction and internal reorganization based upon anticipated changes in the external environment. This situation is exemplified by multiple satellites operating in a cooperative mode. In the event of a satellite failure, remaining satellites would detect autonomously the condition (task deduction) and may generate and execute orbit-and spacecraft-reconfiguration commands.

Level 10. Includes Level 9 and is capable of internal reorganization and dynamic task deduction based on unspecified and unknown/unanticipated changes in external environment. The system will strive to maximize system utility. Thus, mission objectives should be adaptive and automatically reprogrammable. System resources should be maximized to preserve task adaptiveness.

APPENDIX B

SUMMARY OF JPL EXPERIENCE IN ON-BOARD COMPUTING VS. AUTONOMY FOR VIKING AND VOYAGER

Contributors:

D. J. Eisenman
C. P. Jones
E. C. Litty
H. B. Phillips

SECTION 1

VIKING EXPERIENCE

1.1 VIKING MISSION DESCRIPTION

The Viking Mission delivered two spacecraft (S/C) each comprised of an orbiter and probe, into Mars orbit, and performed probe landing and orbital operations. Launch was in the fall of 1975 and the prime mission terminated in mid-November 1976. Extended orbiter operations terminated in July of 1978 and 1980 for orbiter 2 and 1, respectively. Each orbiter contained two programmable subsystems; Computer Command Subsystem (CCS) and Flight Data Subsystem (FDS).

1.2 VIKING COMPUTER SYSTEM DESCRIPTION

1.2.1 CCS

The CCS performed the following functions:

- (1) Decode ground commands.
- (2) Issue discrete and serial data commands to all S/C subsystems.
- (3) Execute CCS processor and memory load commands.
- (4) Store event sequences into CCS memory.
- (5) Command event sequences from CCS memory.
- (6) Output CCS telemetry to the FDS.
- (7) Respond to 32 interrupt pulses and 32 bi-level state change inputs.
- (8) Internally generate timing interrupts to drive software timing routines.
- (9) Internally generate error interrupts to signal CCS hardware and software anomalies.

The CCS was a special purpose digital computer with block redundant elements which were always active. A functional block diagram of the CCS is shown in Figure B-1. The CCS processor was interrupt driven and had 64 instructions. The CCS memory was 4K by 18 bit plated wire, 2K of which was write protected. Fixed routines for command decoding and failure detection and correction were typical of the functions located in write-protected memory. The remaining half of the memory was used to load sequences which controlled the spacecraft's engineering and science subsystems during trajectory correction maneuvers, science data acquisition and transmittal, and various calibration exercises.

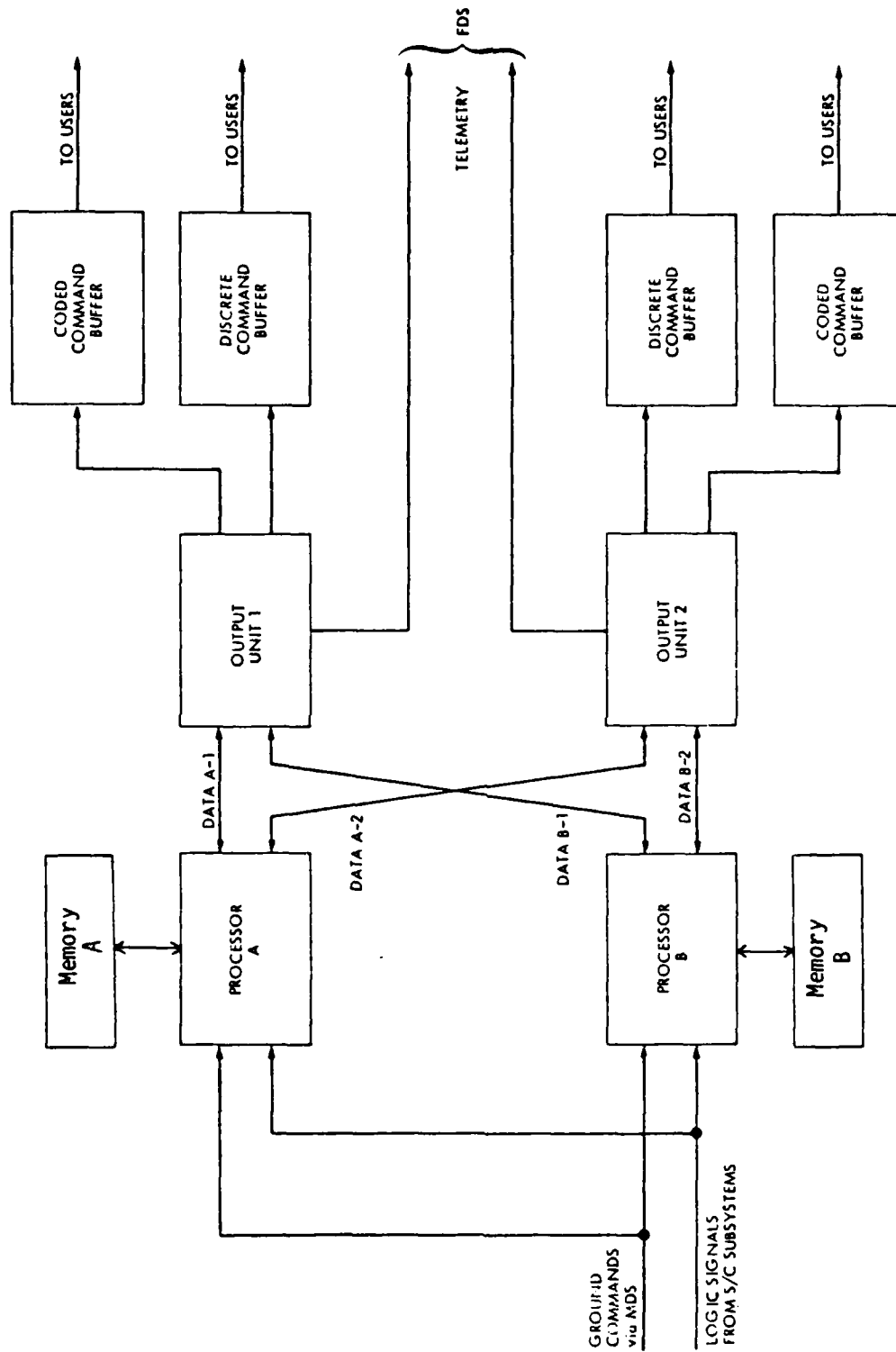


Figure R-1. Viking Computer Command Subsystem (VCS) Block Diagram

1.2.2 FDS

The FDS performed the following functions.

- (1) Collect data from all S/C subsystems.
- (2) Perform analog-to-digital conversions.
- (3) Combine and format engineering and science payload telemetry.
- (4) Control science payload operation.

Interrupt and bi-level state change inputs from FDS to CCS allowed the transfer of low-rate engineering telemetry data. This capability was used in the extended prime mission. The FDS contained a 1K by 8 bit, functionally redundant, plated wire memory which could be modified via CCS commands. The memory contained four different programmable telemetry sampling formats which could be individually selected. A different part of the memory could be loaded and used to sequence science payload operations. Another part of memory was used to buffer science payload and engineering telemetry.

1.2.3 CCS Interfaces

Many S/C subsystems interfaced with the CCS via interrupts and level changes.

1.2.3.1 Propulsion Subsystem (PROP)

- (1) Level input to indicate that the helium pressurant tank regulation had failed.

1.2.3.2 Attitude Control Subsystem (ACS)

- (1) Accelerometer pulse interrupt to indicate an increase in S/C velocity.
- (2) Attitude Control Electronics (ACE) power change level to indicate a probable failure to control S/C attitude.
- (3) Sun acquire level to indicate that the sun sensor was locked onto the sun.
- (4) Star acquire level to indicate that the roll reference star sensor was locked on a star.

1.2.3.3 Power Subsystem (PWR)

- (1) Share mode pulsed level to indicate that boost converter was attempting to remove PWR from an undesirable solar panel battery share operating mode, which could deplete the batteries.
- (2) Two battery high-temperature interrupts to indicate the battery temperature had risen to a point during a charge cycle where damage might result.

1.2.3.4 Radio Frequency Subsystem (RFS)

- (1) Low exciter power level to indicate that the RF power supplied from the exciter to the TWTAs was below tolerance.
- (2) Low TWTAs power level to indicate that the RF power output of the TWTAs was below tolerance.

1.2.4 CCS Software Routine Structure

The routine structure of the CCS had five essential parts, and is depicted in Figure B-2.

- (1) Hardware received levels and timing interrupts from other subsystems on the spacecraft.
- (2) Software preprocessed this data as input.
- (3) Software performed intermediate processing.
- (4) Software generated commands to other subsystems and telemetry as output.
- (5) Hardware generated switch closures or data patterns to other subsystems on the spacecraft.

When a timing or level interrupt occurred, an element of sequence code (e.g., a command to be issued to another subsystem) or a fixed routine executed. Following execution, the software returned to a "wait" state.

1.3 VIKING FAULT PROTECTION ROUTINES

Viking fault protection routines are summarized by prime and extended prime mission. All prime mission routines were used for both orbiter 1 and 2 and were resident in both CCS memories. Not all extended prime mission routines were utilized for both orbiters. Generally, routines used in extended prime mission were resident in only one CCS memory. Some routines in prime and extended prime mission were resident for only part of the mission.

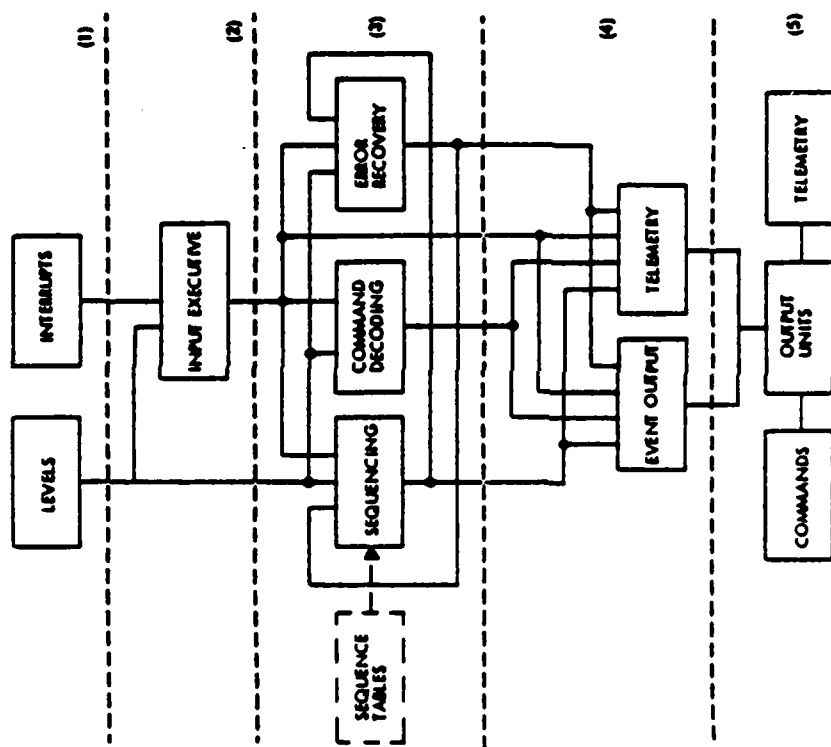


Figure B-2. Viking CCS Software Routine Structure

1.3.1 Prime Mission Routines

Table B-1 summarizes the prime mission fault protection routines:

TABLE B-1

Viking Prime Mission Fault Routine Summary

<u>Routine</u>	<u>Number of CCS Words</u>	<u>% of One CCS Memory</u>
ERROR	379	9.3
ACEPWR	33	.8
BCHGDS	28	.7
CMDLOS	65	1.6
PRSREG	15	.4
RFLOSS	30	.7
ROLREF	34	.8
SHRMOD	70	1.7
SUNACQ	26	.6
MOIMAU	121	3.0
VLSEPI	17	.4
TOTAL	818 words	20%

1.3.1.1 **ERROR.** This routine responded to anomalous CCS hardware and software conditions. The routine response was to save hardware/software status indicators, terminate all ongoing hardware/software activities, reinitialize both hardware and software to a known safe state, and initiate other failure protection routines if selected.

1.3.1.2 **ACEPWR.** This routine responded to an Attitude Control Electronics (ACE) power switch input or was indirectly entered through the SUNACQ failure routine described below. The routine response was to switch in and initialize the standby block redundant ACE or transfer to the ERROR routine if in the Mars Orbit Insertion (MOI) mission phase.

1.3.1.4 **BCHGDS.** The battery charger disconnect routine responded to either of two battery over-temperature interrupt signals. The routine response was to power off the battery chargers.

1.3.1.4 **CMDLOS.** The command loss routine responded to the failure to receive a ground command when a preset number of hours had elapsed since the last received ground command. The routine response selected a known, safe 'downlink' configuration and then preceded to cycle all combinations of primary and secondary receivers, exciters and transmitters until a ground command was received.

1.3.1.5 PRSREG. The pressurant regulator routine responded to an over-pressure monitor input. The routine response was to close a valve on the helium pressurant line to the fuel and oxidizer tanks.

1.3.1.6 RFLOSS. The radio frequency loss routine responded to an input which indicated a low power output condition from the Radio Frequency Subsystem (RFS) exciter and/or traveling wave tube amplifier (TWTA). The routine response, after the low power condition had existed for a set period of time, was to switch in a standby block redundant exciter and/or TWTA.

1.3.1.7 ROLREF. The roll reference loss routine responded to an input from the ACS which indicated the celestial roll reference star was not acquired. The routine response (if the sun was acquired) was to command the ACS to search the entire star tracker field of view (CST flyback) for the reference star until the star was acquired.

1.3.1.8 SHRMOD. The share mode routine responded to a failure condition when the solar panels and batteries were operating together to provide spacecraft power for an excessive period. The routine monitored the number of boost converter pulses necessary to exit this sharing mode. After a preset number did not cause exit of the share mode, spacecraft loads were then shed in sequential pairs.

1.3.1.9 SUNACQ. The sun acquisition routine responded to a sun loss input from the ACS. The routine response, after the sun loss condition had existed for a set period of time, was to enter the ERROR and ACEPWR routines described above.

1.3.1.10 MOIMAU. The Mars Orbiter Insertion (MOI) maneuver routine was executed (via the ERROR routine) in response to a spacecraft power transient or an ACE power switch input. The routine response was to restart the MOI maneuver activity (after being terminated by the ERROR routine), initialize the ACE to a known safe state and to perform the ACE standby block redundant switch, if that was the reason for entering the routine.

1.3.1.11 VLSEPI. The Viking Lander Separation Inhibit routine responded to the sun loss and ACE power switch conditions. This was done by preempting the normal SUNACQ and ACEPWR routine responses. The routine response was to execute commands to inhibit lander separation and then to allow normal ACEPWR and SUNACQ responses.

1.3.2 Extended Prime Mission Routines.

Table B-2 summarizes the extended prime mission fault protection routines.

TABLE B-2

Viking Extended Prime Mission Fault Routine Summary

<u>Routine</u>	<u>Number of CCS Words</u>	<u>% of Total CCS Memory[†]</u>
DECOM	293	3.6
BATMON	20*	.2
BATCHG	113	1.4
RCVRSW	50	.6
CORKER	263	3.2
SINPOM	25*	.3
LEAKCK	50*	.6
STRAY	182	2.2
DLOFF	123	1.5
ACLMON	24	.3
AUTMON	28	.3
TOTAL	1171	14.3%

*Estimates.

[†]Assumes residence in only one CCS Memory.

1.3.2.1 DECOM. The Decommutator Executive routine executed when initialized to read, format and store low rate engineering telemetry data input to the CCS by the FDS. A maximum of 17 telemetry users could be serviced. After all telemetry data was accumulated, each user was given control to process the telemetry data as necessary. Different Viking user routines are described below.

1.3.2.2 BATMON. The battery monitor routine provided protection against battery failure during long (4 hours) occultations by using two battery discharge current telemetry measurements (one/battery), determining an intolerable current unbalance, and then reducing spacecraft power loads in a similar fashion to SHRMOD above.

1.3.2.3 BATCHG. This routine automatically charged the batteries to optimize battery power and extend useful life of the batteries. Outside of occultation periods, both batteries were put into high rate charge. Both battery temperature telemetry measurements were monitored to determine the optimum charging period. Each battery was individually put into the low rate charge when this occurred.

1.3.2.4 RCVRSW. The receiver switch routine provided protection against a receiver failure during extended occultation by monitoring the selected receiver oscillation current telemetry, determining an intolerable current level, and switching to the backup receiver.

1.3.2.5 CORKER. This routine provided protection against attitude control jet gas leaks with a minimum of hands-on operation. This was done by monitoring attitude control deadband telemetry measurements, detecting excursions in pitch, yaw or roll axis and firing the correct jet to attempt to clear the leak.

1.3.2.6 SINPOM. The science instrument power-on monitor routine provided protection against instrument damage by monitoring turn-on current telemetry measurements, determining the presence of intolerable current levels, and individually turning off the 'out-of-spec' instrument.

1.3.2.7 LEAKCK. This routine provided for hands-off exit and entry into a roll-drift spacecraft attitude mode to conserve control gas. Once initiated, this routine would exit the roll drift mode and reacquire the roll celestial reference star. Control gas jet leaks were then monitored and cleared in a similar fashion to CORKER, above, but only on the roll axis. When no leaks were detected, the roll-drift mode was reentered.

1.3.2.8 STRAY. The stray light routine provided protection against star tracker damage and maintained reference star acquisition. This was done by monitoring the star tracker intensity telemetry measurements, determining an intolerable intensity level and then using gyro control for the roll axis and turning the star tracker off. The star tracker was then periodically cycled on/off to determine if the star intensity level returned to a tolerable limit. If so, the star tracker was then turned on and star roll reference was acquired.

1.3.2.9 DLOFF. The down-link off routine provided assurance for end-of-mission radio silence. This was done by monitoring telemetry for a selected low level of attitude control gas. Once this was encountered and the sun reference was lost, both X and S-band transmitters were turned off.

1.3.2.10 ACLMON. The accelerometer monitor routine was in support of an end-of-mission propulsion engine performance evaluation test. It did not use DECOM supplied telemetry measurements but used special purpose accelerometer interrupt inputs. This routine integrated the accelerometer input, detected a 20% decrease in acceleration and terminated the engine burn.

1.3.2.11 AUTMON. The automatic response monitor routine provided performance visibility and audit trails of the failure protection and active DECOM user routines by providing additional indicators in the downlink CCS telemetry.

SECTION 2

VOYAGER (VGR) EXPERIENCE

2.1 VGR MISSION DESCRIPTION

In August and September of 1977, two Voyager spacecraft were launched on four-year-long missions to investigate Jupiter and Saturn, their many satellites, and the traversed interplanetary environment. Voyager 2 is targeted by navigators to eventually rendezvous with Uranus, four years after its encounter with Saturn in August 1981. The planetary encounter phases are each 100 days long and are marked by a 30-day "observatory" phase during which regular, periodic observations are made of the planetary system. The next 30 days, or "far-encounter" phase, included increased observations of the planets' satellites and spacecraft reorientation maneuvers for the purpose of calibrating the various fields and particles instruments. The "near-encounter" phase, typically five days in length, provides the most intense data gathering during the encounter. Experiments utilizing Sun and Earth occultations by the planet are conducted as well as high-resolution observations by the spacecraft's remote sensing instruments. A 30-day "post-encounter" phase follows during which the activity pace drops to that of the earlier far-encounter phase.

Successful encounters with Jupiter (both S/C) and Saturn (Voyager 1) have been accomplished. Between encounters, each spacecraft conducts the necessary calibration exercises to ready itself for the next encounter while the "cruise science" instruments (typically fields and particles) gather information about the interplanetary medium.

2.2 VGR SYSTEM DESCRIPTION

The Voyager spacecraft design is a product of (1) the early (pre-1970) Thermoelectric Outer Planets Spacecraft (TOPS) concept characterized by substantial redundancy and a Self-Test and Repair (STAR) computer; (2) hard fiscal constraints of the 1970's; and, to some extent (3) the recognition that earlier Mariner and Viking-class spacecraft designs, while not boasting the autonomy or operational flexibility of the TOPS design, could, in fact, meet the mission requirements provided that concerns about their long lifetime reliability could be set aside.

Fault-tolerance, as a characteristic of the spacecraft system design, came about as a result of top-level design requirements on the system that were intended to (1) assure maximum fault-tolerance during mission-critical activities (during post-launch injection, at planetary closest approach, during off-Earth point maneuvers, etc.); (2) provide spacecraft safing in response to faults during unattended (non-tracked) cruise; and (3) minimize the required ground support in the event of an on-board fault. The requirements and their implementation had a profound effect on the spacecraft's hardware configuration and software design.

Each S/C contained three programmable subsystems: Computer Command Subsystem (CCS), Flight Data Subsystem (FDS) and Attitude and Articulation Control Subsystem (AACs).

2.2.1 CCS

The CCS performed a function very similar to that performed on Viking. Hardware modifications were made to alleviate problems encountered by Viking, to reconfigure and add subsystem interfaces (the most notable of which was with AACs), and to add protection against the expected radiation environment of Jupiter and Saturn. Though many hardware interfaces had changed, the CCS software routine structure remained very similar to Viking.

2.2.2 FDS

The FDS performed a function similar to that performed on Viking. However, the FDS was a special purpose digital computer with standby, block redundant elements. The FDS memory was 8K by 16 bit CMOS random access memory which could be write-protected in 4K blocks. The FDS processor was driven by a single 2.5 millisecond timing interrupt and had 36 basic instructions. Multi-bit serial data was input for engineering and science payload telemetry and output for science payload control using input/output and direct memory access hardware. Low-rate engineering telemetry data could be input to the CCS in a fashion similar to Viking. This capability was not used during the prime mission. The FDS memories were programmed differently for each mission phase to provide optimum telemetry and science payload control.

2.2.3 AACs

The AACs contained an embedded special purpose, digital computer with standby, block redundant elements. This computer was the processing and control element of the AACs. It performed the following functions:

- (1) Input signals from inertial and celestial sensors and actuators.
- (2) Process (1) via programmed control law algorithms to produce a) torque control for trajectory correction and attitude control thrusters and b) control drive for scan platform actuators.
- (3) Provide FDS with telemetry data to assess performance and status.
- (4) Accept CCS commands to control operating modes.
- (5) Provide 6 level 'power code' inputs to CCS for actions AACs could not accomplish, such as redundant element power switching.

2.2.4 CCS Interfaces

Many other S/C subsystems interfaced with the CCS via interrupt and level changes.

2.2.4.1 Power Subsystem (PWR)

- (1) Undervoltage level to indicate an out-of-tolerance power condition.
- (2) Main to standby inverter switch level which indicated subsequent action taken by PWR in response to a continued undervoltage condition.

2.2.4.2 Radio Frequency Subsystem (RFS)

- (1) One low S-band and one low X-band exciter power level to indicate the RF power supplied from the S, X-band exciters to the S, X-band TWTAs was below tolerance.
- (2) One low S-band and one low X-band TWA power level to indicate that the RF power output of the S, X-band TWA was below tolerance.

2.2.4.3 Infrared Interferometer Spectrometer and Radio Subsystem (IRIS)

- (a) No power level to indicate the failure of the optics heater power supply.

2.3 VOYAGER FAULT PROTECTION

The Voyager fault-protection software exists within two subsystems; the CCS and the AACS. In the former, fault routines are initiated by interrupts received from external sources, and followed by the preprogrammed response. In AACS, however, fault routines are periodically executed and are always comparing current performance indicators against preprogrammed "norms." When an unfavorable comparison occurs, action is taken.

2.3.1 Voyager CCS Fault Protection Routines

Voyager fault protection routines are summarized for the prime mission only. All routines were utilized for both Voyager 1 and 2 and were resident in both CCS memories. The TRNSUP routine was the only one resident for some parts of the mission and not resident for others. Table B-3 summarizes the Voyager CCS fault protection routines.

TABLE B-3

Voyager CCS Fault Routine Summary

<u>Routine</u>	<u>Number of CCS Words</u>	<u>% of One CCS Memory</u>
ERROR	230	5.6
AACSIN	339	8.3
CMDLOS	101	2.5
IRSPWR	20	.5
PWRCHK	167	4.1
RFLOSS	93	2.3
DMLOAD	67	1.6
TRNSUP	68	1.7
TOTAL	1085	26.5%

2.3.1.1 ERROR. This routine was similar in nature to the Viking ERROR routine. The Voyager ERROR routine was not normally entered by other failure protection routines as was the case in Viking. PWRCHK (described below) is the only failure protection routine initiated by ERROR.

2.3.1.2 AACSIN. This routine responded to informational and functional AACS power codes (p.c.'s)(requests for action). Functional p.c. responses resulted in the powering on or off of AACS elements such as; thruster isolation valves, star tracker, star tracker sun shutters, gyros and replacement heaters. Information p.c. responses varied but included: failure protection for celestial reference status change (sun or star loss and acquisition), switching of failed AACS block redundant units, maneuver abort, scan platform slew inhibit, AACS/CCS interface problems, and command sequence errors.

2.3.1.3 CMDLOS. This routine was similar in nature to the Viking CMDLOS routine. The Voyager response provided complete reconfiguration of the antenna-receiver-command detector hardware string. To preclude the loss of command ability due to a false lock condition (receiver locked on a downlink spur), all possible downlink strings were selected for each possible exciter-transmitter configuration. The response executed alternatively using prime and secondary power relay selections and continued until a valid command was received.

2.3.1.4 IRSPWR. This routine provided protection against the failure of the IRIS (Infrared Interferometer Spectrometer and Radio Subsystem) optics heater power supply, which remained on to prevent irreparable damage to the instrument due to improper thermal control. The routine response switched in the block redundant optics heater power supply when the prime unit failed.

2.3.1.5 PWRCHK. The power recovery routine provided protection against three possible abnormal power conditions (two from PWR, one from CCS) by reconfiguring the spacecraft power load into a known, safe, low power mode. The routine provided for a preselected minimum or optimum power level state and also selected redundant units, where possible, to help eliminate the overload.

2.3.1.6 RFLOSS. This routine provided protection against an S or X-band exciter or transmitter output power failure by selecting the appropriate, block redundant, standby units.

2.3.1.7 DMLOAD. This routine was never used in flight but was resident in both memories on both spacecraft. The routine provided protection against memory faults which could prevent commanding a CCS. The routine made use of a CCS-to-CCS hardware interface capability to directly load memory from one CCS to the other.

2.3.1.8 TRNSUP. The tandem and turn command support routine provided protection against CCS and AACS abnormal operations during critical spacecraft maneuvers. The routine response, when an abnormal condition was encountered, was to terminate any ongoing maneuver and then perform a celestial (sun, star) reacquisition.

2.3.2 Voyager AACS Fault Protection

Due to the length of the mission and the long two-way communication times, the AACS was designed to be redundant. A functional block diagram of the AACS is shown in Fig. B-3. The spacecraft is 3-axis stabilized using monopropellant thrusters (0.2 lbf thrust) to provide control torques, and obtains pitch and yaw reference data from wide-angle sun sensors (SS) and roll reference data from the Canopus Star Tracker (CST). Block redundant sun sensors and Canopus Star Trackers are provided. Electrical biases to the sun sensor cause the spacecraft pitch and yaw axes to be offset from the sun, pointing the -Z axis and the high gain antenna at Earth. A dry-gyro inertial reference unit (DRIRU) provides attitude information when celestial references are unavailable (during maneuvers, etc.) with redundant measurements possible for each axis.

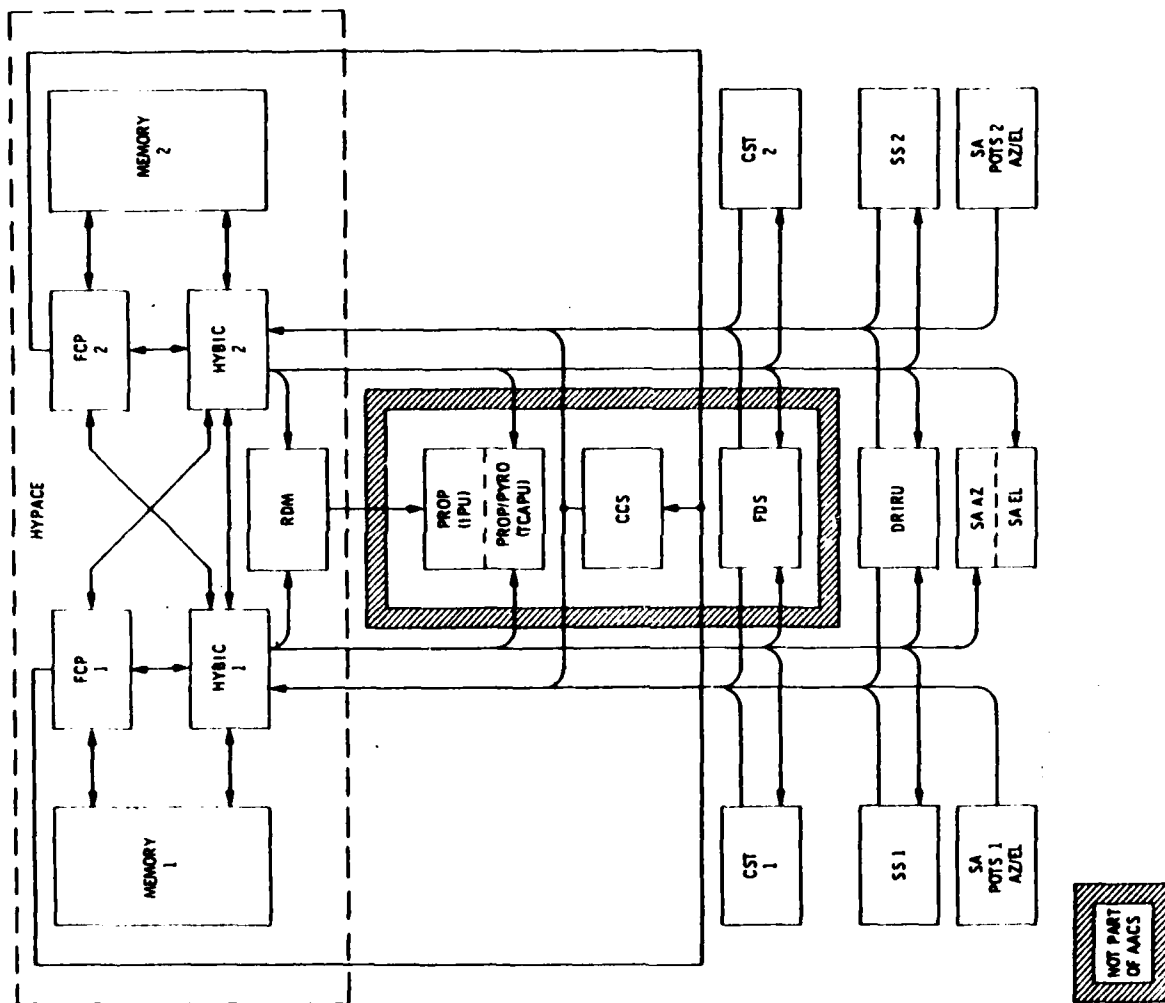


Figure R-3. AACS Functional Block Diagram

18 bit, fixed point, flight control processors (FCP) and two plated-wire memories (MEM), with 4096 words each, provide computational capability. These interface with the remainder of the AACS sensors and actuators through two hybrid interface circuits (HYBIC). While the memories are dedicated to their respective FCP's, either FCP/MEM combination may function with either HYBIC. The HYBIC's also provide interfaces with the trajectory correction and attitude propulsion unit (TCAPU), the flight data subsystem (FDS) computer, and the CCS computer. The term HYPACE (hybrid programmable attitude control electronics) refers to the above components, 2 FCP/MEM's and 2 HYBIC's; and the remote driver module (RDM) described below. Also shown in Figure B-1 are the two redundant sets of scan platform position feedback potentiometers (SA POTS AZ/EL). The scan platform actuators are not redundant, however.

The TCAPU provides a set of non-redundant thrusters for trajectory correction maneuvers (TCM). Either half of the TCM thruster set (pitch) or (yaw) may be used alone, providing a measure of functional redundancy. Two separate sets of attitude control thrusters are provided, identified as branch 1 and branch 2 in Figure 4-17. The IPU module indicated in Figure B-3 refers to the Injection Propulsion Unit thrusters on the Propulsion Module (PM), used for the initial trajectory injection and jettisoned shortly after launch. The remote driver module (RDM) interfaced between the HYBIC and the IPU, providing drive signals to the IPU thrusters.

The Voyager flight software design was very heavily impacted by the limited memory space. Extensive effort and ingenuity was required to perform the necessary functions in the available space. The flight software was written in assembly code. A flow chart of the AACS flight software is shown in Figure 4-14 in the main body of this report. Normal program execution occurs in three different rate groups having periods of 10 ms, 60 ms, and 240 ms. The fourth rate group shown (20 ms) was used only for the Propulsion Module operation. Functions requiring high rates such as thruster activation and scan platform stepper motor operations are performed by the 10 ms logic. The bulk of the attitude control functions, such as attitude sensor 'reads' and control law algorithms, are accomplished by the 60 ms logic. The 240 ms logic performs a variety of tasks that do not require the higher execution rates, such as decoding CCS commands from the input buffer, fault monitor and correction, and "power code" processing.

A "power code" is a 6 bit message sent to the CCS computer, which may be only informational or may cause a power command to switch power to an AACS component. Such power switching commands are usually the means by which redundant elements are exchanged. These power codes are an important part of the fault protection logic, allowing the CCS computer to issue commands in response to a fault condition. These commands may be a simple power command (A gyro on) or a command sequence which will turn the spacecraft in a pattern designed to re-orient the spacecraft towards the sun from an entirely random attitude. Some serious faults result in an OMEN power code, which causes CCS to save the next three power codes (normally lost) for later analysis.

A 10 ms interrupt clock provides the basic timing for the various rate groups. External interrupts will occur from the CCS, presenting a command to AACS to be put into a buffer for decoding during the next 240 ms cycle, or from FDS, presenting a request for AACS telemetry to be provided immediately. Error interrupts may also occur due to HYBIC power interruptions or internal processor errors. All interrupts, except the FCP internal error interrupt, may be disabled when necessary, such as during sensor 'read' tasks.

The fault monitor and correction block contains fault routines which are summarized in Table B-4.

TABLE B-4

Summary of Voyager AACS Fault Protection Routines

<u>Routine</u>	<u>Number of AACS Words</u>	<u>% of One AACS Memory</u>
DRIRU	165	4.0
TCAPUF	95	2.4
Power Supply Monitor	22	0.6
Plated Wire Refresh	27	0.6
FCP Test Control	19	0.4
CCS Comm. Interpreter	35	0.8
Celestial Sensor Logic	135	3.2
Power Code Processor	64	1.6
Catastrophe Handler	32	0.8
Miscellaneous Other Functions	200	4.8
TOTAL	794	19.4%

2.3.2.1 DRIRU Test Routine. Each pair of the three orthogonally mounted gyros have one axis in common. Since the gyros are normally powered in pairs, the outputs from the two gyros for the common axis may be compared as an error check. In the event of a failure, this routine will cycle through the three possible combinations of the three gyros. If none of the three combinations function, a HYBIC replacement (swap) is requested, assuming a problem in the interface circuits. The use of DRIRU information is inhibited during the warm-up phase of the gyros, and the test disabled during periods when the gyros are not functional (such as celestial cruise).

2.3.2.2 TCAPU Fault Correction Control. This routine tests for thruster failures by monitoring the number of thruster pulses used in each five minute period, and the spacecraft attitude error. Thus, thrusters that fail in either the open or closed condition will be detected. If a failure occurs, the appropriate thrusters will be replaced by the alternate branch. If this fails to clear the problem, an interface problem is assumed and a HYBIC swap is requested. If the possible combinations of thrusters and HYBIC's fail to clear the problem, an FCP swap is requested. TCAPU fault testing is disabled during the start and end of turns due to the large number of pulses and position overshoot which is normal at this time.

2.3.2.3 Power Supply Monitor. This routine monitors a weighted average of two power supply voltages, causing a HYBIC swap if an abnormal indication persists.

2.3.2.4 Plated Wire Refresh. This routine refreshes the plated wire memory (one location every 30.7 seconds) and compares the refreshed contents with the original. If a discrepancy occurs, the FCP/MEM is assumed to be bad and an FCP swap is requested.

2.3.2.5 FCP Self Test Control. The FCP has an internal hardware self-test which may be initiated via an external interrupt. This routine is used for this self-test.

In addition to the above, a significant amount of the fault detection and correction logic is distributed in other routines in the program. Three of these are described below:

2.3.2.6 CCS Command Interpreter. Two checks are made to detect bit errors, (one a parity check) and the commands are checked for proper order, where appropriate. For example, data must always precede a command requiring separate data, and a turn command should never occur when the spacecraft is not in an inertial mode. A significant amount of logic is devoted to determining if commands "make sense" at the time they are received. Certain critical commands are required to be preceded by a 'precursor' command.

2.3.2.7 Celestial Sensor Logic. The celestial sensor logic evaluates the information from the sun sensor and Canopus star tracker, and provides the necessary control of the Canopus star tracker operating modes. This logic also controls the sequencing of the spacecraft state from the All Axes Inertial mode, where gyro references are used, to the Celestial Cruise mode, which uses the sun and Canopus (or another star) as attitude references. The state of the spacecraft and its progression through the acquisition sequence is constantly monitored, enabling the proper corrective action to be taken in the event of an anomalous indication.

Possible corrective measures include a HYBIC swap, which brings the alternate celestial sensors into use, or a return back to an inertially referenced mode. In one failure mode, complete loss of the sun reference, a power code is sent to CCS, which initiates a series of pitch and yaw turns which result in a 4 steradian search for the sun with the sun search flag enabled. The celestial sensor logic can then terminate the turns when the sun has been reacquired. This CCS routine is called the IDET (read IDET not) routine, which stands for Illumination Detection. The IDET routine has proven useful in several cases where no actual failure occurred, but a sun search enable was commanded at a time when the sun was not in the sun sensor field of view due to errors in the sequence of commands sent to the spacecraft.

2.3.2.8 Power Code Processor Routine. The power code processor routine handles the transmission of the six bit power codes to the CCS computer. At the point in the logic where it is determined that power code(s) should be sent, the power code(s) are loaded into a queue that can hold up to 18 power codes. The power code processor routine then unloads this queue, at a rate of one power code per 240 ms, sending the current power code to the power code register in the FCP. This creates an interrupt to the CCS computer, which then reads the contents of the register, and acts upon the contents. A command containing the same power code (called an echo power code command) is sent back to AACS. The CCS generally responds only once per second, limiting the ability of AACS to "tie up" CCS with power code traffic.

If the echoed power code matches the power code originally sent, then the power code processor proceeds to send the next power code in the queue. If no echo occurs, or the wrong echo occurs, the power code processor waits for 7.2 seconds and then initiates corrective action, which may include a HYBIC swap or FCP swap. If the power code queue has been emptied, AACS sends a special "heart beat" power code (one per 1.92 seconds). The CCS uses the presence of power code traffic to assess the health of the AACS FCP. The AACS, in turn, can cause the CCS to swap FCP's, by stopping this "heart beat" (and all other power codes).

2.3.2.9 Catastrophe Handler. This routine is called by the power code processor. It either:

- (1) Swaps HYBIC, or
- (2) If HYBIC has been swapped, it kills the heart beat and induces a processor swap.